# A UNIFYING FRAMEWORK FOR MODEL REDUCTION BY LEAST-SQUARES PADÉ APPROXIMATION

**Ian David Smith**

A Thesis submitted in partial fulfilment
of the requirements of the
University of Abertay Dundee
for the degree of Doctor of Philosophy

*April 1998*

**I certify that this thesis is the true and accurate version of the thesis approved by the examiners.**

Signed ..... [redacted] ..... Date....13/8/98
***Director of Studies***

## ACKNOWLEDGEMENTS

I wish to thank my director of studies Dr T N Lucas for his unfailing guidance and support throughout this project, my second supervisors, Dr R Paris and Mr A Milne, and the Division of Mathematical Sciences for providing the research funds required.

I am very grateful to Professor I C Colligan, recently retired Head of the School of Informatics, for first suggesting the possibility of a research project within the School.

Finally, I must thank the Founding Committee of Muscat College of Management Science and Technology in the Sultanate of Oman and its Chairman, Mr Ahmed Al-Ghazali, for granting permission to use the College's facilities to complete this thesis.

## ABSTRACT

A thorough review of the literature on the model reduction of linear, time-invariant, dynamical systems in both the frequency and time domains is presented. Particular attention is paid to the least-squares extension of the classical method of Padé approximation. An account is given of the development of apparently different approaches of least-squares parameter-matching Padé model reduction applied to continuous-time and discrete-time systems. These approaches are shown to be related via a unifying theory. From the formulation it is possible to show several interesting features of the least-squares approach which lead to a fuller understanding of exactly how the reduced model approximates the full system. An error index is derived in the general continuous-time case and it is shown that a range of system parameter preservation options are available. Using the theory developed in the continuous-time case a non-uniqueness property of the method is proven. An 'optimal' least-squares method based on the approach and the introduction of weighting for the system parameters are both investigated. The unifying theory is extended to the discrete-time case where an important new stability preservation property is proved and is shown to provide the basis for a new least-squares Padé method. This method uses transformations between the $z$- and $s$-planes to guarantee stable reduced order models approximating stable high order continuous-time systems. The application of least-squares Padé approximation is further extended to the multivariable case with particular attention given to the factors affecting the levels of order reduction achieved. Appropriate numerical examples are used to illustrate the main points of the thesis and graphs of the impulse and step responses are used throughout to visually highlight the accuracy of approximation.

## AIMS/OBJECTIVES

- Review the literature on the model reduction problem, especially in the frequency domain
- Investigate, in particular, the least-squares Padé approximation techniques proposed to-date
- Develop software to facilitate the gathering of empirical evidence about the model reduction of a large number of discrete-time examples
- Develop a thorough and well-founded mathematical understanding of least-squares Padé approximation
- Establish the nature of the relationship between the various least-squares methods
- Assess the performance of the least-squares methods of model reduction in comparison with other frequency domain methods
- Investigate an apparent stability preservation property in the discrete-time case
- Investigate the possibility of a new stability-preserving method for continuous-time systems
- Extend the application of least-squares model reduction to the multivariable case
- Examine the nature of any difficulties affecting the extension of the method to the simplification of multivariable systems

## CONTENTS

# CHAPTER 1

# INTRODUCTION

## 1.1 Order Reduction of Linear Dynamical Systems

Simplification is an essential starting point in the analysis, design and control of real systems. Indeed, the bulk of systems dealt with by scientists and engineers are non-linear and infinite-dimensional and so some simplifying assumptions *must* be made. The most common assumption made is that, at a particular operating point, it is possible to represent the system by a linear, finite-dimensional model.

However, even this simplification may not be enough in itself as the linear dynamical model proposed is often described by systems of high-order differential equations, or, alternatively, by transfer functions of high order. Therefore, the approximation of such high order plant and control systems has long been a part of control system design. In the past engineers have achieved this approximation by applying intuitive, physically based assumptions (Green and Limebeer 1995). Unfortunately, physical intuition can often be misleading and result in expensive penalties in financial and safety terms. For this reason there has been much interest over the past 35 years in the development of mathematical procedures known as *model reduction techniques.*

This additional simplification of systems by the application of mathematical methods is desirable for the following reasons (Towill 1972):

- It renders the system under investigation more readily understandable
- It reduces computational requirements for ease and cheapness of simulation and/or real-time application
- It reduces incidence of human error
- It increases the reliability of prediction from restricted system data
- It simplifies the generalisation of results collected for particular systems.

Further, the continued interest and large number of methods available in the literature testify to the importance of obtaining reliable, simplified models for the analysis and design of systems in general.

## 1.2 The Model Reduction Problem

The focus of this research project is on model reduction in the frequency domain as opposed to the time domain. However, an understanding of model reduction in the time domain is important for a full appreciation of the issues raised by the model reduction problem. Accordingly, an account is given here of both the transfer function and state-space representations of linear systems. Also, linear dynamical systems are characterised for mathematical analysis by dynamic equations. These may specify the relationships between the rates of change of time-varying quantities (continuous-time), or they may specify the relationships between values of the quantities at specific points in time (discrete-time). Since this project examines the model reduction of both continuous-time and discrete-time systems the account of basic concepts given here will be for continuous-time systems with appropriate comments, where necessary, relating to equivalent results for discrete-time systems.

**Single Input/Single Output Systems**

For a continuous-time, linear, time-invariant, dynamical system having input $u(t)$ and output $y(t)$ the dynamic equation is given by (Jacobs 1993)

$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \ldots + a_1 \frac{dy}{dt} + a_0 y = b_{n-1} \frac{d^{n-1} u}{dt^{n-1}} + b_{n-2} \frac{d^{n-2} u}{dt^{n-2}} + \ldots + b_1 \frac{du}{dt} + b_0 u \qquad (1.1)$$

This equation specifies the relationships between the rates of change. This basic time domain representation of the general SISO system may be expressed instead in the frequency domain by the application of the Laplace transform $L$ defined over $t \geq 0$ by

$$L\{y(t)\} = Y(s) \equiv \int_0^\infty y(t) e^{-st} dt \qquad \text{the inverse transform given by} \qquad L^{-1}\{Y(s)\} = y(t)$$

It is well known (Jacobs 1993) that the application of this definition and integration by parts gives the result

$$L\left\{\frac{dy}{dt}\right\} = \int_0^\infty \frac{dy}{dt} e^{-st} dt = sY(s) - y(0)$$

When this is applied repeatedly to (1.1), assuming zero initial conditions, the transform of this equation is

$$\left(a_n s^n + \ldots + a_0\right) Y(s) = \left(b_{n-1} s^{n-1} + \ldots + b_0\right) U(s)$$

where $U(s) = L\{u(t)\}$.

The frequency domain representation called the "transfer function" of the original system is defined from this as the ratio

$$G(s) \equiv \frac{Y(s)}{U(s)} = \frac{b_{n-1}s^{n-1} + \ldots + b_1 s + b_0}{a_n s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0}$$

In a similar fashion, the $z$-transform may be applied to the $n$th order difference equation (Jacobs 1993)

$$\begin{aligned} a_n y(n) + a_{n-1} y(n-1) + \ldots + a_0 y_0 \\ = b_{n-1} u(n-1) + b_{n-2} u(n-2) + \ldots + b_0 u_0 \end{aligned} \qquad (1.2)$$

describing the dynamics of the general discrete-time system. Using the application of the $z$-transform $Z$ defined for a *causal* sequence $\{x_k\}$ for $k = 0, 1, \ldots, \infty$

$$Z\{x_k\}_0^\infty = X(z) = \sum_{k=0}^{\infty} \frac{x_k}{z^k} \quad \textit{the inverse transform given by} \quad Z^{-1}\{X(z)\} = \{x_k\}_0^\infty$$

this description provides the definition of the transfer function in the discrete-time case

$$G(z) \equiv \frac{Y(z)}{U(z)} = \frac{b_{n-1}z^{n-1} + \ldots + b_1 z + b_0}{a_n z^n + a_{n-1}z^{n-1} + \ldots + a_1 z + a_0}$$

where $Y(z) = Z\{y(t)\}$ and $U(z) = Z\{u(t)\}$.

Therefore, in the frequency domain model reduction may be defined as the approximation of the high order system represented by $G(s)$ in continuous-time or $G(z)$ in discrete-time by the reduced order system represented by the transfer function

$$R(s) = \frac{d_{k-1}s^{k-1} + \cdots + d_1 s + d_0}{s^k + e_{k-1}s^{k-1} + \cdots + e_1 s + e_0} \qquad k < n$$

or $$R(z) = \frac{d_k z^k + \cdots + d_1 z + d_0}{z^k + e_{k-1} z^{k-1} + \cdots + e_1 z + e_0} \qquad k < n$$

respectively. It is this definition of model reduction that is used in the project to explore its central interest - the deeper mathematical understanding of specific frequency domain model reduction techniques.

An alternative approach to the description of the general continuous-time linear system with dynamic equation (1.1) is to rewrite this single *n*th order differential equation as a system of *n* first order differential equations. For illustration consider the system with dynamic equation

$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \ldots + a_1 \frac{dy}{dt} + a_0 y = u$$

which may be rewritten as a system of *n* first order equations by defining the variables

$$x_1 \equiv y, \quad x_2 \equiv \frac{dy}{dt}, \quad \ldots \quad x_n \equiv \frac{d^{n-1} y}{dt^{n-1}}$$

giving

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
&\vdots \\
\dot{x}_n &= -\frac{a_{n-1}}{a_n} x_n - \ldots - \frac{a_0}{a_n} x_1 + \frac{1}{a_n} u
\end{aligned}$$

where the output is $y = x_1$.

Such systems of equations may be written in matrix-vector form

$$\begin{aligned}
\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + Bu(t) \\
y(t) &= C\mathbf{x}(t) + Du(t)
\end{aligned} \qquad (1.3)$$

This is known as the *state-space representation* as it defines the system on an $n$-dimensional space using the $n$ state variables $x_i$ ($i = 1, 2, \ldots, n$). Every point in the state-space may be regarded as an initial condition associated with a trajectory representing a solution of (1.3).

In the discrete-time case, a similar rewriting of the difference equation (1.2) results in the state-space representation

$$\begin{aligned} \mathbf{x}(i+1) &= A\mathbf{x}(i) + Bu(i) \\ y(i) &= C\mathbf{x}(i) + Du(i) \end{aligned} \qquad (1.4)$$

$$\text{where } \mathbf{x}(i) = \begin{bmatrix} x_1(i) & x_2(i) & \cdots & x_n(i) \end{bmatrix}^T$$

Therefore, in the time domain, model reduction may be defined as the approximation of an $n$th order system represented by $\{A, B, C, D\}$ in (1.3) or (1.4) by a $k$th order system $\{\overline{A}, \overline{B}, \overline{C}, \overline{D}\}$ where $k < n$. The reduction in the dimension of the state-vector indicates that, in state-space terms, model reduction involves neglecting or "condensing" certain states of the full system. This is discussed more fully in chapter 2.

**Multivariable systems**

One of the main reasons for the importance of the state-space representation of linear dynamical systems is that the restriction to single input/single output systems implicit in the transfer function representation does not apply. The state equations given in (1.3) and (1.4) can describe the dynamics of systems with many inputs and outputs for both the continuous-time and the discrete-time cases. The matrices $A$, $B$, $C$ and $D$ can provide a complete specification of any linear system.

Considering for simplicity only the continuous-time case, a linear, time-invariant dynamical system with $n$ state variables, $m$ input and $l$ output variables may be described in the time domain by the equations

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \qquad (1.5)$$

$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t)$$

where

$$\mathbf{x}(t) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \mathbf{u}(t) = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \qquad \mathbf{y}(t) = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_l \end{bmatrix}$$

and $A$, $B$, $C$ and $D$ are $n$ x $n$, $n$ x $m$, $l$ x $n$ and $l$ x $m$ respectively. Therefore, the transition from SISO to multivariable systems in this representation consists only of taking values other than unity for $l$ and $m$. This means that the definition of model reduction in the time domain is exactly the same for the multivariable case as for SISO systems. On the other hand, there is no natural definition of multivariable model reduction, in frequency domain terms. However, it is of interest to note that the transfer function matrix can be obtained easily from the state-space representation in both the continuous-time and discrete-time cases (Bultheel and Van Barel 1986) using

$$G(s) = C[sI - A]^{-1}B$$

with the complex variable $s$ simply replaced by $z$ for discrete-time, $D$ being assumed null because this is the case for any *proper* transfer function without loss of generality.

## 1.3 Errors in Model Reduction

For model reduction as for any other approximation process there is a need to decide upon an appropriate method for measuring the error introduced by adopting the reduced model as a replacement for the full model. It is seen (section 1.2) that, in the frequency domain, the reduction problem is of one finding a low order $R(s)$ to approximate the high order transfer function $G(s)$ so that the full and reduced systems yield similar responses to the same input, i.e.,

$$L^{-1}\{R(s)U(s)\} \approx L^{-1}\{G(s)U(s)\}$$

where $L^{-1}$ is the inverse Laplace transform.

**System Responses**

When considering the accuracy of approximation of a reduced model the input functions used most frequently by authors are the impulse and step functions. This is because they are mathematically convenient and good accuracy obtained for these inputs generally implies similar levels of accuracy for other inputs as they are also fairly robust basis functions.

The impulse or Dirac delta function is denoted by

$$u(t) = \delta(t)$$

which has Laplace transform of unity, i.e.

$$L\{u(t)\} = U(s) = 1$$

Hence, given the definition of the full system response

$$y(t) \equiv L^{-1}\{G(s)U(s)\}$$

we have the impulse response given as

$$L^{-1}\{G(s)\} \text{ (full model)} \qquad \text{and} \qquad L^{-1}\{R(s)\} \text{ (reduced model)}$$

The unit step function is defined by

$$u(t) = 1 \qquad t > 0$$
$$u(t) = 0 \qquad t < 0$$

for which the Laplace transform is

$$L\{u(t)\} = U(s) = \frac{1}{s}.$$

Therefore, for the step response we obtain

$$L^{-1}\left\{\frac{G(s)}{s}\right\} \text{ (full model)} \qquad \text{and} \qquad L^{-1}\left\{\frac{R(s)}{s}\right\} \text{ (reduced model).}$$

It is common to use graphs of these responses to give a straightforward means of comparison between the full and reduced order models. Such graphs will be used as appropriate in this thesis. However, they provide a purely visual comparison and we need to look also at a quantitative measure of closeness of approximation.

**Integral Square Errors**

The integral square errors of the impulse and step responses as defined above have proved through experience to be a useful measure of how good an approximation has been achieved. They can be calculated quite simply from the transfer function coefficients (Aguirre 1994c) and are defined as

$$\text{Integral Square Error (ISE) } I = \int_0^\infty [y(t) - \bar{y}(t)]^2 dt$$

where $y(t)$ is the full system output and $\bar{y}(t)$ is the reduced system output

If the output functions referred to are impulse responses or step responses this definition will give the Impulse or Step ISE respectively. Such an absolute measure of error still makes a judgement difficult as to how good an approximation a given reduced model happens to be. To facilitate that judgement the Relative ISE may be used instead. This compares the ISE with the total "energy" of the full system and is defined as follows

$$Relative\ ISE = \frac{\int_0^\infty [y(t) - \bar{y}(t)]^2 dt}{\int_0^\infty [y(t) - y(\infty)]^2 dt}$$

The Relative Impulse ISE will be referred to throughout as $I_{rel}$ and the Relative Step ISE as $J_{rel}$. These error indices are provided where appropriate for the reduced models given in examples throughout the thesis.

**Square Error Sum**

All of the above error measures refer to the continuous-time case, but the same general points can be made about accuracy measurement in the discrete-time case where the relevant error indices for the impulse and step responses are summations rather than integrals. For an absolute measure of accuracy the Square Error Sum may be defined as

$$\text{Square Error Sum (SES)} = \sum_{i=0}^{\infty} [y(i) - \bar{y}(i)]^2$$

where $y(i)$ is the full system output and $\bar{y}(i)$ is the reduced system output

The corresponding relative error measure is defined as

$$SES_{rel} = \frac{\sum_{i=0}^{\infty}[y(i) - \bar{y}(i)]^2}{\sum_{i=0}^{\infty}[y(i) - y(\infty)]^2}$$

for the time response. Throughout the thesis where appropriate, these errors will be given for discrete-time reduced models.

## 1.4 Structure of the Thesis

After this introductory chapter there is a concise review of the main trends in time domain model reduction given in Chapter 2 before attention shifts to frequency domain methods which are the main focus of this work. In Chapter 3, a detailed background is provided in the principal developments related to frequency domain model reduction in general. Padé approximation is given particularly careful consideration because of the importance of a good understanding of this classic reduction procedure for the work on the least-squares extension that follows.

Chapter 4 introduces not only the basic idea of least-squares approximation but also reviews all the least-squares Padé methods proposed in the literature over the last decade or so. This review sets the scene for a major contribution of the work presented in this thesis, placing all the least-squares Padé methods within a unifying framework. The development of this framework is presented in Chapter 5 where much material is presented that has been published during the research period (Smith and Lucas 1995, Smith and Lucas 1996, Lucas and Smith 1995).

Chapter 6 extends this framework to the least-squares Padé approximation of discrete-time systems and an important stability preservation result is proven (Lucas

and Smith 1998). The application of the latter to both discrete-time and continuous-time systems is also demonstrated. In Chapter 7, extension of the ideas developed in Chapter 5 to the simplification of multivariable systems is given and, finally, in Chapter 8 the work is placed in the context of the literature on least-squares Padé methods to date and the main conclusions and contributions reviewed.

## 1.5 Model Reduction Software

During the initial research a need was perceived for a computer-based investigation of the application of the least-squares Padé method to the model reduction of discrete-time systems matching Markov parameters only. To facilitate the generation of the required results a QBASIC program was written by the author that performed least-squares model reduction on any discrete-time system using two apparently different methods based on the work of Lalonde *et al* (1992b). A listing of this software is given in Appendix 3.

A large body of results produced by the application of this program provided empirical evidence for two important observations. First, the author could find no case where an unstable reduced order model resulted, suggesting a stability preservation property for which a proof was subsequently forthcoming and is given in Chapter 6 of this thesis. Second, there was no significant difference between the reduced order models produced using the apparently different least-squares methods for any system investigated. It was this latter observation and the attempt to explain it that produced the major contribution of a unifying framework for all least-squares Padé model reduction methods developed extensively in Chapters 5 and 6.

# CHAPTER 2

# MODEL REDUCTION METHODS IN THE TIME DOMAIN

## 2.1 Introduction

It is very common for design engineers to study complex processes via the simple transfer function representation of linear time-invariant systems outlined in section 1.2. However, because of the ease with which the state-space representation may be applied to the multivariable case much work has been carried out on model reduction/simplification in the time domain especially since Davison's (1966) seminal paper on the simplification of linear dynamical systems. Davison proposed one of the first methods for model order reduction, retaining the dominant eigenvalues of the original system. Although these methods are not central to the concerns of this research project, a brief description of the main points of this work will prove useful for an overall understanding of the model reduction problem.

All of the main model reduction/simplification methods in the time domain are based on the principle of truncation. This principle seeks to remove from the state-space model (Green and Limebeer 1995) a number of states that are 'unimportant' to the behaviour of the system in some sense to be defined. For instance, we might take 'unimportant states' to mean those states associated with eigenvalues that have large, negative, real parts. Different definitions of differing mathematical subtlety may be adopted while the basic principle applied remains the same. In the following sections, the main time domain techniques and the relationships between them will be described.

## 2.2 The Truncation Method

The state-space representation of an $n$th order system is given by the equations

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \qquad (2.1)$$
$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t)$$

as discussed in chapter 1. To apply truncation to this system it is necessary to divide the state vector $\mathbf{x}(t)$ into the components for retention and those for removal so that

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix}$$

where for reduction to a $k$th order model the vector $\mathbf{x}_2(t)$ contains the components of the state vector which are to be regarded as 'unimportant'. A partitioned form of (2.1) would be

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\mathbf{x}(t) + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}\mathbf{u}(t) \qquad (2.2)$$
$$\mathbf{y}(t) = \begin{bmatrix} C_1 & C_2 \end{bmatrix}\mathbf{x}(t) + D\mathbf{u}(t)$$

The $k$th order system is derived by simply removing the components of $\mathbf{x}_2(t)$ to obtain

$$\dot{\mathbf{z}}(t) = A_{11}\mathbf{z}(t) + B_1\mathbf{u}(t) \qquad (2.3)$$
$$\mathbf{w}(t) = C_1\mathbf{z}(t) + D\mathbf{u}(t)$$

where $\mathbf{z}(t) \equiv \mathbf{x}_1(t)$ and $\mathbf{w}(t) \cong \mathbf{y}(t)$. This basic approach produces a $k$th order model about which little is known except that it will match perfectly at infinite frequency, i.e.

$$\mathbf{R}(\infty) = \mathbf{G}(\infty)$$

where $\mathbf{R}(s)$ and $\mathbf{G}(s)$ are the transfer function matrices associated with the truncated and full systems respectively.

The method is known as modal truncation when the truncation process described above is applied to a modal realization of the full system, that is where the matrix $A$ in (2.1) is in the form

$$A = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

where the $\lambda_i$ ($i = 1, 2, ..., n$) are the eigenvalues of the system, assumed to be distinct for simplicity. Because of the form of $A$ the characteristics of the reduced order model yielded by truncation may be varied by ordering the eigenvalues in ascending order of magnitude of $Re(\lambda_i)$. It is of interest to note also that the slowest modes are not always dominant (as Aguirre (1994b) points out) because the error associated with the removal of a particular mode depends, not on $|Re(\lambda_i)|$ alone, but on the ratio

$$\frac{\|\mathbf{G}_i\|}{|\mathrm{Re}(\lambda_i)|}$$

Therefore, the error is affected also by the size of the residues $\mathbf{G}_i$ which may be defined in terms of $\mathbf{G}(s)$ in the following fashion (Sinha 1984)

$$\mathbf{G}(s) = \sum_{i=1}^{n} \frac{\mathbf{G}_i}{(s - \lambda_i)} \qquad \text{where} \qquad \mathbf{G}_i = \lim_{s \to \lambda_i} (s - \lambda_i)\mathbf{G}(s)$$

This method has proved popular because of its simplicity in conceptual and in computational terms and the guarantee of stability given by the fact that the poles of the reduced order system are a subset of the poles of the full system. This last point makes this method the time domain analogue of the frequency domain technique of pole retention described in section 3.3.

## 2.3 The Perturbation Method

Despite its attractions state-space truncation has a steady-state step response error given by (Green and Limebeer 1995)

$$C_1 A_{11}^{-1} B_1 - C A^{-1} B$$

which can be large in cases where good accuracy at low frequencies is required. This problem may be overcome by the method of singular perturbation which also uses the partitioned form (2.2) of the state equations. However, in this case, the 'fast' modes of the system are not simply ignored, but their behaviour approximated by making the assumption that

$$\dot{\mathbf{x}}_2(t) = 0$$

This assumption gives

$$\mathbf{0} = A_{21}\mathbf{x}_1(t) + A_{22}\mathbf{x}_2(t) + B_2\mathbf{u}(t)$$

which is used to substitute for $\mathbf{x}_2(t)$ in (2.2) giving a $k$th order approximation with the state equations

where

$$\dot{\mathbf{z}}(t) = \hat{A}_{11}\mathbf{z}(t) + \hat{B}_1\mathbf{u}(t) \qquad (2.4)$$

$$\mathbf{w}(t) = \hat{C}_1\mathbf{z}(t) + \hat{D}\mathbf{u}(t)$$

$$\hat{A}_{11} = A_{11} - A_{12}A_{22}^{-1}A_{21} \qquad \hat{B}_1 = B_1 - A_{12}A_{22}^{-1}B_2$$

$$\hat{C}_1 = C_1 - C_2A_{22}^{-1}A_{21} \qquad \hat{D} = D - C_2A_{22}^{-1}B_2$$

It can be shown (Green and Limebeer 1995) that this procedure is equivalent to performing the frequency domain transformation

$$\mathbf{H}(s) = \mathbf{G}\left(\frac{1}{s}\right)$$

before carrying out a state-space truncation on $\mathbf{H}(s)$ to obtain $\mathbf{T}(s)$ and taking $\mathbf{R}(s)$ to be the result of a reciprocal transformation applied to $\mathbf{T}(s)$, i.e.,

$$\mathbf{R}(s) = \mathbf{T}\left(\frac{1}{s}\right)$$

The fact that singular perturbation is related to the truncation method in this way means that the perfect matching of $\mathbf{T}(s)$ and $\mathbf{H}(s)$ at infinite frequency for state-space truncation implies zero steady-state error for singular perturbation approximation.

## 2.3 Balanced Truncation and Perturbation

It is seen that the properties exhibited by a reduced order model obtained by state-space truncation will depend on the realization of the system that is selected for reduction. The time domain method of balanced truncation (Moore 1981) uses this fact to identify a particular type of system realization which, if selected, will give good absolute error results as given by the $H_\infty$-norm defined as the maximum error between $\mathbf{G}(s)$ and $\mathbf{R}(s)$ for the range 0 to infinity

$$\|\mathbf{G} - \mathbf{R}\|_\infty$$

Analysis suggests (Green and Limebeer 1995) that in order to keep this error small the appropriate realization for truncation is a *balanced realization* $(A, B, C)$. This realization is asymptotically stable and satisfies the conditions

$$A\Sigma + \Sigma A^T + BB^T = 0$$
$$A^T\Sigma + \Sigma A + C^TC = 0$$

where

$$\Sigma = \begin{bmatrix} \sigma_1 I_{r_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \sigma_2 I_{r_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \sigma_m I_{r_m} \end{bmatrix} \qquad \left(\sigma_i \neq \sigma_j, i \neq j, \sigma_i > 0\right)$$

$r_i$ $(i = 1, 2, \ldots, m)$ is the multiplicity of $\sigma_i$ and the $I_r$ are $r_i \times r_i$ identity matrices.

A truncation performed on such a realization is called a balanced truncation if $\Sigma$ is partitioned as

$$\Sigma = \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \Sigma_2 \end{bmatrix}$$

where the partitions $\Sigma_1$ and $\Sigma_2$ are such that no states corresponding to any $\sigma_i$ with multiplicity greater than 1 is split between them. The procedure consists of partitioning the realization $(A, B, C)$ as in (2.2) conformably with $\Sigma$ so that the truncated system is as given in (2.3) such that its state-space is spanned by the eigenvectors corresponding to the larger eigenvalues of

$$P^{\frac{1}{2}}QP^{\frac{1}{2}}$$

where $P$ and $Q$ are controllability and observability gramians respectively.

It can also be shown (Green and Limebeer 1995) that the reduced order model obtained by balanced truncation is itself a balanced realization. Therefore, this time domain method guarantees the stability of the reduced model given the asymptotically stable nature of the full system as well as the minimality of the realization obtained for the reduced order model.

As in the case of straight state-space truncation, balanced truncation tends not to produce good approximations in the low-frequency region. To overcome this problem in cases where this is important the same solution of adopting the method of singular perturbation may be applied. Hence, the method of balanced singular perturbation is equivalent to the frequency domain transformation

$$\mathbf{H}(s) = \mathbf{G}\left(\frac{1}{s}\right)$$

applied to a balanced realization of the full system, followed by letting $\mathbf{T}(s)$ be the $k$th order truncation of $\mathbf{H}(s)$ and then performing the reciprocal transformation

$$\mathbf{R}(s) = \mathbf{T}\left(\frac{1}{s}\right)$$

Again the realization produced by this procedure will be stable and minimal, but gives a better approximation at low frequencies than balanced truncation, and has zero steady-state error. Examples of the good results achieved by these methods may be found in the literature (Moore 1981, Pernebo and Silverman 1982, Fernando and Nicholson 1983).

## 2.4 Summary

The above sections give a concise overview of the most popular model reduction or simplification techniques developed for the time domain. It was seen that there are two main requirements for a reduction or simplification method to be useful. Firstly, it should preserve stability, i.e., the reduced order model is always stable if the full system is stable and, secondly, that it should give a low absolute approximation error for the frequency range important for the design problem being addressed. In the chapters that follow these requirements will prove to be central for the assessment of model reduction schemes whether in the time or the frequency domain.

# CHAPTER 3

# MODEL REDUCTION METHODS IN THE FREQUENCY DOMAIN

## 3.1 Introduction

Many and varied model reduction techniques have been developed in the frequency domain. These have enjoyed widespread popularity with the engineers who are familiar with the ideas and methods of classical control theory upon which these methods are based. In particular, this is true in relation to work with high order SISO systems because the system description is usually identified in transfer function form. These methods form the background against which the development of the idea of least-squares Padé approximation, which is the focus of this thesis, must be set. Therefore, a description of the main frequency domain methods will help clearly define the issues that will be explored later in chapters 4-7.

The structure of the chapter follows the chronology of the development of these methods, starting with the continued-fraction and Padé methods (section 3.2) which were the first major attack on the model reduction problem in the frequency domain. Problems of accuracy and 'stability preservation' were immediately apparent in relation to the continued-fraction/Padé techniques. Therefore, during the 1970's, there was much work on the so-called 'stability preserving' methods (section 3.3) which is an important strand in the literature of control system design. However, a close look is also taken in section 3.2 at the ways in which the research on the continued-fraction/Padé methods has resulted in an improvement in the 'robustness' of the techniques, allowing for a greater flexibility of application to overcome the problems mentioned above.

## 3.2 Continued-Fraction/Padé Model Reduction Methods

Here, consideration is given to an extremely important class of model reduction methods which have proven popular since their first proposal in the late 1960's and early 1970's (e.g., Chen and Shieh 1968, Chuang 1970, Shamash 1973c). The methods of continued-fraction (C-F) expansion and Padé approximation, although equivalent in the sense of producing the same reduced order model, have essential differences in their execution.

**Continued-Fraction Expansion Method**

The derivation of reduced-order models by the truncation of a C-F expansion of the transfer function $G(s)$ was pioneered by Chen and Shieh (1968), who use a Cauer-type C-F expansion for this purpose (Wall 1948). When the method was first proposed as 'a novel approach' it attracted a great deal of attention as a method of great simplicity and elegance despite the serious drawback that a stable reduced model could not be guaranteed.

Consider the $n$th-order system transfer function given in usual form by

$$G(s) = \frac{b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \cdots + b_0}{a_n s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \cdots + a_0} \tag{3.1}$$

which is expanded in the C-F 2nd Cauer form as

$$G(s) = \cfrac{1}{h_1 + \cfrac{s}{h_2 + \cfrac{s}{h_3 + \cfrac{s}{h_4 + \cfrac{s}{\ddots + \cfrac{s}{h_{2n}}}}}}} \tag{3.2}$$

Where the continued-fraction expansion is started by dividing the denominator polynomial of $G(s)$ by the numerator polynomial from constant terms, thus giving

$$G(s) = \frac{1}{h_1 + sH_1(s)} \tag{3.3}$$

where $h_1 = a_0 / b_0$ and $H_1(s)$ is a rational function with numerator and denominator degrees equal to $(n-1)$. The process is then repeated on the rational function $H_1(s)$ to give

$$H_1(s) = \frac{1}{h_2 + H_2(s)}$$

where $h_2$ is a constant and $H_2(s)$ is a rational function with numerator and denominator degrees of $(n-2)$ and $(n-1)$ respectively. This process may be continued for $2n$ divisions to obtain the C-F expansion in (3.2). A reduced $k$th order model may then be obtained by truncating (3.2) at the $2k$th convergent

$$R(s) = \cfrac{1}{h_1 + \cfrac{s}{h_2 + \cfrac{s}{\ddots + \cfrac{s}{h_{2k}}}}}$$

which may be inverted to give

$$R(s) = \frac{d_{k-1}s^{k-1} + \cdots + d_1 s + d_0}{e_k s^k + e_{k-1}s^{k-1} + \cdots + e_1 s + e_0} \tag{3.4}$$

which may or may not be stable for a given stable $G(s)$.

Chen and Shieh (1968) show that the $h_i$ ($i = 1, 2, \ldots, 2n$) in (3.2) may be easily calculated by a Routh-type array algorithm of the form

$$\begin{array}{ccccc} A_{11} & A_{12} & \cdots & \cdots & A_{1,n+1} \\ A_{21} & A_{22} & \cdots & A_{2,n} & \\ A_{31} & A_{32} & \cdots & A_{3,n} & \\ \vdots & \vdots & & \vdots & \\ A_{2n,1} & & & & \\ A_{2n+1,1} & & & & \end{array} \tag{3.5}$$

where the first row consists of the denominator cofficients and the second row consists of the numerator coefficients of the transfer function both from the constant terms, i.e., $A_{1,i} = a_{i-1}$ and $A_{2,i} = b_{i-1}$, $i = 1, 2, \ldots, n+1$ $(b_n = 0)$. The remaining elements in the array are calculated using the normal Routh algorithm

$$A_{i,j} = A_{i-2,j+1} - h_{i-2} A_{i-1,j+1} \quad i = 3, 4, \ldots, 2n+1, \; j = 1, 2, \ldots, n+1-[i/2]$$

and

$$h_i = A_{i,1} / A_{i+1,1}$$

where $[\phi]$ means the integer part of $\phi$. It is noticed that inversion of the reduced model's C-F expansion can also be efficiently derived by a table similar to (3.5) and that the resulting reduced model has the property of retaining the first $2k$ terms in the series expansion of $G(s)$ about $s = 0$.

**Example 3.1**

To illustrate this basic technique consider the seventh-order system given by the transfer function

$$G(s) = \frac{1441.53s^3 + 78319s^2 + 525286.125s + 607693.25}{s^7 + 112.04s^6 + 3755.92s^5 + 39736.73s^4 + 363650.56s^3 + 759894.19s^2 + 683656.25s + 617497.375}$$

This practical example is taken from Chen and Shieh (1968) and gives the following C-F expansion of the transfer function by (3.5)

$$G(s) = \cfrac{1}{1.016 + \cfrac{s}{4.054 + \cfrac{s}{-0.067 + \cfrac{s}{-3.804 + \cfrac{s}{-11.04 + \ddots}}}}}$$

Simplifying to a second-order reduced model by retaining the first four quotients

and discarding the remainder gives

$$R(s) = \cfrac{1}{1.016 + \cfrac{s}{4.054 + \cfrac{s}{-0.067 + \cfrac{s}{-3.804}}}}$$

which inverts to

$$R(s) = \frac{0.25s + 1.03}{s^2 + 0.51s + 1.05}$$

with

$$I_{rel} = 3.0384\% \quad \text{and} \quad J_{rel} = 1.5731\%$$

The simplicity of the method together with the fact that it gives good stable approximations in many cases has led to efforts of refinement to improve still further the accuracy of the reduced models (considered in detail in the next subsection). At this stage, however, it is noted that stability preservation is a *serious* problem for the basic C-F expansion method. This is illustrated by the following stable fourth-order transfer function

$$G(s) = \frac{9s^3 + 42s^2 + 31s + 10}{s^4 + 8s^3 + 21s^2 + 22s + 8}$$

which on reduction to second-order by the method of Chen and Shieh (1968) gives the unstable model

$$R(s) = \frac{-2.9224s - 0.4636}{s^2 - 2.2081s - 0.3709}$$

The basic C-F idea has also been extended to the reduction of discrete-time systems by Shamash (1974). In this paper, Shamash notes that C.F. expansion about $s = 0$ is equivalent to C.F. expansion about $z = 1$ in the discrete-time case. To take

account of this a linear transformation $z = p+1$ is used, thus enabling application of the basic C-F method to the transformed transfer function $H(p) = G(p + 1)$. Truncation to a $k$th order model after inverse transformation gives

$$R(z) = \cfrac{1}{h_1 + \cfrac{(z-1)}{h_2 + \cfrac{(z-1)}{h_3 + \ddots + \cfrac{(z-1)}{h_{2k}}}}}$$

which may be inverted in the usual way via a Routh-type array. Shamash also shows how the initial terms in the expansion about $z = \infty$ of the discrete-time system may be retained to derive a biased reduced order model in a manner similar to that for the continuous-time models.

**Generalised Continued-Fraction Expansion Method**

In the previous section, it was acknowledged that the C-F model reduction technique of Chen and Shieh (1968) can exhibit problems of accuracy of approximation to the full system response and, more seriously, of the possible generation of unstable reduced order models. However, efforts have been made to refine and modify this simple model reduction method in order to overcome these problems to some extent. These have proved successful in maintaining the position of the C-F methods as among the most popular and widely used approaches in model reduction.

Being equivalent to Padé approximation about the point $s = 0$, the technique of Chen and Shieh (1968), when it produces stable models, tends to ensure good steady-state (low frequency) matching of the full and reduced systems. However, Chuang

(1970) observed that it may sometimes produce a poor approximation to the transient response of the full system, which is largely governed by the high frequency parameters obtained by expansion about $s = \infty$. For this reason Chuang (1970) proposed a different form of C-F expansion to include information about the full high-order system at $s = \infty$ to produce a 'biased' reduced order model, which claimed to give a good approximation of both steady-state and initial transient response of the full system. This is accomplished by alternately dividing the rational function parts, $H_1(s)$, $H_2(s)$, etc. in (3.3), of the continued-fraction from lowest and highest powers of $s$ respectively at each division stage.

By this means a mixed C-F expansion is produced of the form

$$G(s) = \cfrac{1}{h_1 + \cfrac{s}{h_2 + \cfrac{1}{h_3 + \cfrac{s}{h_4 + \cfrac{1}{\ddots + \cfrac{s}{h_{2n}}}}}}} \qquad (3.6)$$

where the $h_i$ are the low frequency parameters for odd $i$ and the high frequency parameters for even $i$. A reduced $k$th order model may then be obtained by truncating the C-F expansion (3.6) at the $h_{2k}$ parameter. This gives rise to a model with an equal number ($k$) of series expansion terms matched to $G(s)$ when expanded about $s = 0$ (time moments) and $s = \infty$ (Markov parameters) respectively. This reduced model will then tend to reflect both steady-state and transient characteristics of the full system.

Lucas (1983b) proposed an algorithm which generalises the method of Chuang (1970) for producing biased models. It provides a systematic way of calculating the $h_i$ for the expansion which forms the first $t$ quotients by division from lowest powers

and the next $(2k - t)$ quotients by division from highest powers. This simple and effective modification of Chuang's method means that the combinations of time moments and Markov parameters retained may be varied to perhaps further improve the accuracy of the reduced order model.

A number of authors further developed the ideas of biased C-F reduction methods e.g., Davidson and Lucas (1974), Parthasarathy and Jayasimha (1982), Pal (1986), Lucas (1983c) and (1983d). These involved C-F expansions using a general (real) expansion point $s = a$ as well as the points at $s = 0$ and $s = \infty$. Lucas (1986) builds on his previous work of two-point C-F expansions to argue forcefully for the robustness of C-F/Padé methods. As he points out, it is a simple matter to extend the biased model algorithm to three expansion points, with the first $t$ time moments and the first $m$ Markov parameters being retained together with $(2k - t - m)$ terms about $s = a$. However, choice of the general expansion point $a$ needs careful consideration for some systems. It is seen that with a simple modification of the algorithm, a full multipoint C-F expansion may be achieved with up to a maximum of $2k$ distinct points being matched, although in most cases the three points 0, $a$ and $\infty$ are enough for a good approximation. This establishes the C-F expansion approach as a truly powerful and flexible model reduction tool in spite of its lack of guaranteed stability preservation.

The flexibility of the multipoint C-F method is further underlined by the work of Katsube *et al* (1985) and Hwang and Lee (1989), in which the latter authors propose a Jordan-type C-F expansion about the arbitrary frequency points $s = \pm i\omega_j$. The method gives an efficient model reduction algorithm which avoids the complex

arithmetic required by the method of Katsube *et al* (1985). The *k*th order reduced model is given by truncating the C-F expansion

$$G(s) = \cfrac{1}{h_1 + k_1 s + \cfrac{s^2 + \omega_1^2}{h_2 + k_2 s + \cfrac{s^2 + \omega_2^2}{h_3 + k_3 s + \cfrac{s^2 + \omega_3^2}{\ddots \\ h_{n-1} + k_{n-1} s + \cfrac{s^2 + \omega_{n-1}^2}{h_n + k_n s}}}}}$$

after the first $k$ partial quotient pairs, $(h_j,k_j), j = 1, 2, \ldots, k$. The expansion points $\pm i\omega_j$ on the imaginary axis may be chosen to match the frequency characteristics of the full system at these points.

**Classical Padé Approximation Method**

Very closely related to the continued-fraction expansion method of the previous section, and of fundamental importance to the development of frequency domain model reduction methods, is the method of Padé approximation. The method in its simplest form (Shamash 1975a) consists of matching the Taylor series expansions about $s = 0$ of the high-order system transfer function $G(s)$ and the reduced order model $R(s)$ for as many terms as possible.

The definition of the $\{m, k\}$ Padé approximant of the transfer function $G(s)$ is the rational function

$$R(s) = \frac{N_m(s)}{D_k(s)}$$

where $N_m(s)$ and $D_k(s)$ are polynomials of degree $m$ and $k$ respectively $(m < k)$ and the Taylor series expansion (about $s = 0$) of $R(s)$ is identically equal to that of $G(s)$ up to, and including, the term in $s^{m+k}$.

In what follows, $m$ is taken to be equal to $(k - 1)$ (since this retains the maximum number of system parameters) without loss of generality and $R(s) = \{k-1,\ k\}$ will be the $k$th order Padé approximant of $G(s)$. Therefore, if we consider the usual $n$th order transfer function given in (3.1) which may be expressed in the Taylor series expansion form

$$G(s) = c_0 + c_1 s + c_2 s^2 + \ldots \tag{3.7}$$

where the $c_i$ are the time moment "proportionals" (hereafter simply referred to as the time moments) of the $n$th order system and are defined by matching

$$G(s) = \int_0^\infty e^{-st} g(t) dt$$

to (3.7) giving

$$c_i = \frac{(-1)^i}{i!} \int_0^\infty t^i g(t) dt \qquad i = 0, 1, 2, \ldots$$

Taking the $k$th order Padé approximant of $G(s)$ to be

$$R(s) = \frac{d_{k-1} s^{k-1} + \cdots + d_1 s + d_0}{s^k + e_{k-1} s^{k-1} + \cdots + e_1 s + e_0}$$

and equating to the Taylor expansion for $G(s)$ in (3.7) up to, and including, the term in $s^{2k-1}$, a set of $2k$ linear equations is obtained, i.e.

$$\begin{aligned}
d_0 &= e_0 c_0 \\
d_1 &= e_0 c_1 + e_1 c_0 \\
&\vdots \\
d_{k-1} &= e_0 c_{k-1} + e_1 c_{k-2} + \cdots + e_{k-1} c_0 \\
0 &= e_0 c_k + e_1 c_{k-1} + \cdots + e_{k-1} c_1 + c_0 \\
&\vdots \\
0 &= e_0 c_{2k-1} + e_1 c_{2k-2} + \cdots + e_{k-1} c_k + c_{k-1}
\end{aligned} \tag{3.8}$$

The $2k$ unknown coefficients of $R(s)$ are then found by solving equations (3.8). It is noted that the $c_i$ can be determined recursively from the coefficients of the full system's transfer function using the following formulae

$$c_i = \frac{b_i - \sum_{j=0}^{i-1} c_j a_{i-j}}{a_0} \qquad i = 0,\ 1,\ \ldots\ ,\ n-1$$

and

$$c_i = \frac{-\sum_{j=0}^{i-1} c_j a_{i-j}}{a_0} \qquad i = n,\ n+1,\ \ldots \tag{3.9}$$

It is well known that (3.8) can be expressed in the matrix-vector form

$$A\mathbf{x} = \mathbf{b} \tag{3.10}$$

where

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 & -c_0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & -c_1 & -c_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & -c_{k-1} & -c_{k-2} & \cdots & -c_0 \\ 0 & 0 & \cdots & 0 & -c_k & -c_{k-1} & \cdots & -c_1 \\ 0 & 0 & \cdots & 0 & -c_{k+1} & -c_k & \cdots & -c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & -c_{2k-1} & -c_{2k-2} & \cdots & -c_k \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{k-1} \\ e_0 \\ e_1 \\ \vdots \\ e_{k-1} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ c_0 \\ c_1 \\ \vdots \\ c_{k-1} \end{bmatrix}$$

In this form the reduced model's numerator and denominator coefficients are determined by obtaining the solution of equation (3.10) as

$$\mathbf{x} = A^{-1}\mathbf{b}$$

assuming $A$ to be non-singular.

**Example 3.2**

Again consider the seventh-order example of Chen and Shieh (1968) from illustrative example 3.1. The time moments for this system are given as

$$c_0 = 0.9841 \qquad c_1 = -0.2389$$

$$c_2 = -0.8197 \qquad c_3 = 0.6243$$

Therefore, the matrix-vector equation (3.10) to be solved for the second-order Padé approximant is given by

$$\begin{bmatrix} 1 & 0 & -0.9841 & 0 \\ 0 & 1 & 0.2389 & -0.9841 \\ 0 & 0 & 0.8197 & 0.2389 \\ 0 & 0 & -0.6243 & 0.8197 \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ e_0 \\ e_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0.9841 \\ -0.2389 \end{bmatrix}$$

which gives the reduced model

$$R(s) = \frac{0.25s + 1.034}{s^2 + 0.51s + 1.05}$$

This is identical to the reduced model obtained in example 3.1 using Cauer-type C-F expansion.

The simplicity of the method combined with its ready adaptation to matrix methods of calculation makes it an obvious candidate for order reduction of high order systems. It is very economical in computational terms since the number of equations to be solved in (3.8) depends on the order of the *reduced* model required and not on that of the full system. Also it is noted that unlike the C-F method which requires $G(s)$ to derive $R(s)$ the Padé method requires only the time moments of the full system. Finally, although the numerator degree of the approximant can vary between 0 and $k$ - 1, it is usually desirable to retain the maximum number of system parameters ($2k$ time moments) in the reduced model when calculating the $k$th order approximant. For this reason the accuracy of the approximation is often very good when the method does yield a stable reduced model.

**Generalised Padé Approximation Method**

As in the case of the equivalent C-F expansion methods, it is seen in the last section that the classical Padé method is affected by the possible generation of unstable reduced order models and possible poor approximation to the transient response of the full system. A way of overcoming these problems would be to use Padé approximation about two or more points.

Consider the $n$th order transfer function given in (3.1). A power series expansion of $G(s)$ about $s = \infty$ results in

$$G(s) = \frac{m_1}{s} + \frac{m_2}{s^2} + \frac{m_3}{s^3} + \cdots \tag{3.11}$$

where the $m_i$ are the Markov parameters which govern the transient phase of the $n$th order system. They are found by equating powers of $s$ in descending order between (3.1) and (3.11) to give the system of equations

$$\begin{aligned}
b_{n-1} &= m_1 \\
b_{n-2} &= m_1 a_{n-1} + m_2 \\
b_{n-3} &= m_1 a_{n-2} + m_2 a_{n-1} + m_3 \\
&\vdots \\
b_0 &= m_1 a_1 + m_2 a_2 + \cdots + m_{n-1} a_{n-1} + m_n \\
0 &= m_1 a_0 + m_2 a_1 + \cdots + m_n a_{n-1} + m_{n+1} \\
&\vdots \\
0 &= m_n a_0 + m_{n+1} a_1 + \cdots + m_{2n-1} a_{n-1} + m_{2n}
\end{aligned} \tag{3.12}$$

and the usual technique for their calculation is a recursive procedure given by

$$m_j = b_{n-j} - \sum_{i=1}^{j-1} m_i a_{n-j+i} \qquad j = 1, 2, \ldots, n$$

$$m_j = -\sum_{i=j-n}^{j-1} m_i a_{n-j+i} \qquad j = n+1, n+2, \ldots$$

Applying this definition, the exact Padé method for producing biased two point approximations may be developed easily by considering the case where a $k$th order reduced model is derived by matching $k + t$ time moments and $k - t$ Markov parameters $(0 \le t \le k)$ of the full system. In this case, the numerator and denominator coefficients of the reduced model are derived from the following sets of equations

$$\begin{aligned}
d_0 &= e_0 c_0 \\
d_1 &= e_0 c_1 + e_1 c_0 \\
&\vdots \\
d_{k-1} &= e_0 c_{k-1} + e_1 c_{k-2} + \cdots + e_{k-1} c_0 \\
0 &= e_0 c_k + e_1 c_{k-1} + \cdots + e_{k-1} c_1 + c_0 \\
&\vdots \\
0 &= e_0 c_{k+t-1} + e_1 c_{k+t-2} + \cdots + e_{t-1} c_t + c_{t-1}
\end{aligned} \tag{3.13}$$

being the Padé equations for the matched time moments, $c_i$ $(i = 0,\ 1,\ \ldots\ ,\ k + t - 1)$, and

$$\begin{aligned}
d_{k-1} &= m_1 \\
&\vdots \\
d_t &= m_{k-t} + m_{k-t-1} e_{k-1} + \cdots + m_2 e_{t+2} + m_1 e_{t+1}
\end{aligned} \tag{3.14}$$

for the matched Markov parameters, $m_j$ $(j = 1,\ 2,\ \ldots\ ,\ k - t)$. Substituting from (3.14) for the $d_i$ $(i = t, t+1, \ldots, k-1)$ into (3.13) gives the following matrix-vector form for the two point exact Padé method

$$\begin{bmatrix} c_{k+t-1} & c_{k+t-2} & \cdots & \cdots & \cdots & \cdots & c_t \\ c_{k+t-2} & c_{k+t-3} & \cdots & \cdots & \cdots & c_t & c_{t-1} \\ \vdots & \vdots & & & & \vdots & \vdots \\ c_{k-1} & c_{k-2} & \cdots & \cdots & \cdots & c_1 & c_0 \\ c_{k-2} & c_{k-3} & \cdots & \cdots & \cdots & c_0 & -m_1 \\ c_{k-3} & c_{k-4} & \cdots & \cdots & c_0 & -m_1 & -m_2 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ c_t & c_{t-1} & \cdots & c_0 & -m_1 & \cdots & -m_{k-t-1} \end{bmatrix} \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{k-1} \end{bmatrix} = \begin{bmatrix} -c_{t-1} \\ -c_{t-2} \\ \vdots \\ -c_0 \\ m_1 \\ m_2 \\ m_3 \\ \vdots \\ m_{k-t} \end{bmatrix} \quad (3.15)$$

which is a generalisation of (3.10). The numerator coefficients are then calculated by (3.14) and/or (3.13) as appropriate.

However, difficulties arise with the application of the Padé method about two or more points that do not affect the C-F techniques. The difficulties relate to the linear shift, $s = p + a$, which is required to establish a set of Padé equations similar to (3.8) for the general expansion point $s = a$. It is only in the case of multipoint Padé approximation about $2k$ real distinct expansion points that the application of matrix methods of calculation appears possible, as shown by Therapos and Diamessis (1984) and Pal (1986).

Another interesting proposal for increasing the flexibility of the method is the "frequency-fitting" Padé approximation method due to Xiheng (1987). This method effectively tries to improve the classical Padé method by matching selected frequencies near each prominent peak or valley in the full system's frequency response. At the same time, the low-frequency response of the full model is approximated via the usual Padé equations. The method is seen to perform well even when reducing highly oscillatory systems. Unfortunately, the lack of a systematic procedure for the identification of the selected frequencies (Aguirre 1992) means that high computational costs may be incurred.

However, generalised multipoint Padé approximation has recently been shown to rival the generalised C-F methods by Lucas (1993a). In this paper, a flexible and computationally efficient method of formulating a multipoint Padé approximant is proposed. By converting the problem of approximating a rational function into one of polynomial approximation, Lucas reduces the process of multipoint Padé approximation to a Routh-type division algorithm. The process is then seen to be suitable for the application of matrix methods involving no complex arithmetic regardless of whether the $2k$ expansion points are distinct, real, complex or purely imaginary. Lucas (1993b) proposes a highly effective and robust optimal method based on this new matrix method and the same author (1993c) completely generalizes this method for multipoint Padé approximation by including the expansion point at $s=\infty$. This has transformed multipoint Padé approximation into a completely general technique applicable equally to discrete-time and continuous-time systems without many of the computational difficulties previously associated with Padé/C-F methods. The results obtained are seen to compare favourably with those obtained by stability preservation methods.

### 3.3 Stability Preservation Methods

It was seen (section 3.2) that the possibility of producing an unstable reduced model is a *serious* difficulty for the basic C-F/Padé methods. One way of overcoming this difficulty is to use alternative model reduction techniques, the primary function of which is to guarantee the stability of any reduced models produced, given a stable full system. In this section consideration will be given to the main 'stability preservation' techniques which have been developed through this approach to the model reduction problem.

The methods of Pole Retention, Routh Approximation and Stability Equation are described in the following subsections as these appear to be the three main approaches of stability preservation upon which other more refined methods have been based. No attempt is made to identify any of the methods as generally superior to the others, although the same sixth-order transfer function is used as an illustrative example for all three stability preserving techniques.

**The Pole Retention Method**

An important concept of stability preservation in the frequency domain is that of retention of certain poles from the original high order system in the reduced system (Shamash 1974a, 1975a). This idea is the direct frequency domain analogue of the time domain method based on retention of dominant system modes (Davison 1966, Jamshidi 1983, Green and Limebeer 1995) often referred to as the "modal truncation" technique.

Consider the $n$th order transfer function, whose poles are known, given by

$$G(s) = \frac{b_{n-1}s^{n-1} + \cdots + b_1 s + b_0}{(s+p_1)(s+p_2)\ldots(s+p_n)}$$

where $\mathrm{Re}(p_i) > 0 \quad (i = 1,\ 2,\ \ldots\ n)$ with

$$|p_1| \le |p_2| \le |p_3| \le \ \ldots\ \le |p_n|$$

It is clear that for $k < n$ a reduced degree denominator formed by retaining the first $k$ poles of smallest magnitude, $-p_i$, $i = 1,\ 2,\ \ldots\ k$, will result in the poles of the reduced order transfer function also being stable. Notice that complex poles would need to be retained in conjugate pairs for denominator polynomials with real coefficients. The smallest magnitude poles are retained because they tend to influence steady-state responses more so than the largest magnitude poles.

A variety of techniques may be used in conjunction with pole retention to establish the numerator polynomial for a $k$th order model given by

$$R(s) = \frac{d_{k-1}s^{k-1} + \cdots + d_1 s + d_0}{(s+p_1)(s+p_2)\ldots(s+p_k)}$$

Shamash (1975a) outlines a "partial" Padé method, using Padé approximation to calculate the numerator by matching the first $k$ time moments of the full and reduced systems. Lucas (1983a) shows how the method of "Factor Division" may be used to reduce the computational effort and outlines a simple Routh-type algorithm for the calculation of the numerator.

A method also making use of the idea of pole retention is that outlined in Shamash (1975a) where the author uses pole retention to stabilise unstable reduced models which have been produced using straightforward Padé approximation. The method involves the retaining of poles of the full system one at a time from the smallest magnitude upwards until a stable reduced order model is achieved. Also, Lucas (1994) outlines a frequency domain method for producing suboptimal reduced order models, in the sense of minimizing an integral square error index, for a predetermined reduced model's denominator. The method is easy to implement, being a multipoint Padé approximation technique, involving only matrix multiplication and the solution of a set of linear equations

The pole retention method described so far assumes retention of the slowest modes, i.e., viewing the poles closest to the imaginary axis as dominant. However, it should be noted that Aguirre (1994b) argues that in many cases the slowest modes are not dominant. He proposes the use of "modal dominance indices" (MDIs) which measure the dominance of system poles and provides an illustrative example to show that truly dominant poles can be those further away from the imaginary axis.

Clearly, the idea of retaining dominant poles can be translated directly to the discrete-time case where stability needs to be preserved by exactly the same arguments as in the continuous-time case. However, it is well-known (Shamash and Feinmesser 1978) that in discrete-time the steady-state behaviour of a system is largely dictated by the poles nearest the point $z = 1$ in the $z$-plane. Therefore, it would be those poles with $|z|$ closest to 1 in the right half plane that would be retained rather than the poles of smallest magnitude as in the continuous-time case.

Although exceedingly simple in its conception, it is clear that this method shares with the analoguous time domain technique the need to calculate all the poles of the original high order system, as well as clarifying a criterion for identifying the dominant or otherwise desirable poles. Shamash (1975a) addresses the latter issue by outlining a method for identifying dominant poles by the application of Koenig's theorem. Also, in the same paper, he introduces a matrix interpretation of the partial Padé method which indicates that information concerning the neglected poles is included in the computation and that there is exact steady-state matching of the full and reduced systems for polynomial inputs.

**The Routh Approximation Method**

The model reduction method of "Routh approximation" was first proposed by Hutton and Friedland (1975) and has proved to be the most popular and enduring of the stability preserving methods. The method is so-called because it is based upon the stability properties of the well-known Routh table associated with linear systems. A number of modifications on the Routh method have since been proposed in the literature (e.g., Krishnamurthi and Seshadri 1976 and 1978, Shamash 1975b and 1978, Rao *et al* 1978, Sarasu and Parthasarathy 1979, Hwang *et al* 1995), but the method as originally proposed for continuous-time systems is now briefly outlined.

For the Routh method, the rational transfer function

$$G(s) = \frac{b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \cdots + b_0}{a_n s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \cdots + a_0}$$

is expanded in the '*alpha-beta*' continued-fraction form (Hutton and Friedland 1975) which is expressed as

$$G(s) = \sum_{l=1}^{n} \beta_l \prod_{j=1}^{l} F_j(s)$$

where the $\beta_l$ are constants and the $F_j(s)$ $(j > 1)$ are given by the continued fraction expansion

$$F_j(s) = \cfrac{1}{\alpha_j s + \cfrac{1}{\alpha_{j+1}s + \cfrac{1}{\alpha_{j+2}s + \ddots \; \alpha_{n-1}s + \cfrac{1}{\alpha_n s}}}} \qquad (3.16)$$

the leading denominator term being $1 + \alpha_1 s$ rather than $\alpha_1 s$ for $j = 1$.

This continued-fraction expansion is derived by dividing both the numerator and denominator polynomials of $G(s)$ by the alternant polynomial

$$a_{n-1}s^{n-1} + a_{n-3}s^{n-3} + \cdots + a_1 s$$

where $n$ has been assumed even without loss of generality. The result may be expressed as a product of the new numerator and

$$F_1(s) = \frac{1}{1 + \alpha_1 s + F_2(s)}$$

where $$F_2(s) = \frac{1}{\alpha_2 s + F_3(s)}$$

and generally, $$F_i(s) = \frac{1}{\alpha_i s + F_{i+1}(s)} \qquad (i = 2,\ 3,\ \ldots\ ,\ n-2)$$

with $$F_{n-1}(s)=\frac{1}{\alpha_n}$$

where all the $F_i$ $(i=1,\ 2,\ \ldots\ ,\ n-1)$ are derived from the alternant division process.

As an illustration, consider the third order general transfer function

$$G(s)=\frac{b_2s^2+b_1s+b_0}{a_3s^3+a_2s^2+a_1s+a_0}$$

whose $\alpha$-$\beta$ expansion is

$$G(s)=\cfrac{1}{1+\alpha_1 s+\cfrac{1}{\alpha_2 s+\cfrac{1}{\alpha_3 s}}}\left[\beta_1+\cfrac{1}{\alpha_2 s+\cfrac{1}{\alpha_3 s}}\left[\beta_2+\frac{\beta_3}{\alpha_3 s}\right]\right]$$

where the $\alpha_i$ and $\beta_i$ $(i=1,\ 2,\ 3)$ may be obtained from the following $\alpha$- and $\beta$-tables:

<u>$\alpha$-table</u>

$$\begin{array}{lll} \alpha_1=a_3/a_2 & < & a_3 \quad a_1 \\ \alpha_2=a_2/a_1' & < & a_2 \quad a_0 \\ \alpha_3=a_1'/a_0' & < & a_1' \\ & & a_0' \end{array}$$

<u>$\beta$-table</u>

$$\begin{array}{ll} & b_2 \quad b_0 \\ \beta_1=b_2/a_2 & \\ \beta_2=b_1/a_1' & b_1 \\ \beta_3=b_0'/a_0' & b_0' \end{array}$$

where $a_1'=a_1-a_0\alpha_1$, $a_0'=a_0$ and $b_0'=b_0$.

In general, the $\alpha$ parameters are calculated using the Routh stability array. The $\alpha$-table takes the following form:

<u>$\alpha$-table</u>

$$\begin{array}{ccccc} A_{11} & A_{12} & \cdots & \cdots & A_{1,[(n+1)/2]} \\ A_{21} & A_{22} & \cdots & \cdots & A_{2,[(n+1)/2]} \\ A_{31} & A_{32} & \cdots & A_{3,[n/2]} & \\ \vdots & \vdots & & \vdots & \\ A_{n+1,1} & & & & \end{array}$$

where the entries for the first two rows are given by the denominator coefficients of $G(s)$, i.e.

$$A_{1,j} = a_{n-2(j-1)} \text{ and } A_{2,j} = a_{n-(2j-1)}$$

$$j = 1,\ 2,\ \ldots,\ [(n+1)/2]$$

and the remaining entries are obtained from the algorithm

$$A_{i,j} = A_{i-2,j+1} - \alpha_{i-2} A_{i-1,j+1},\quad \alpha_{i-2} = \frac{A_{i-2,1}}{A_{i-1,1}}$$

$$i = 3,\ 4,\ \ldots,\ n+1$$

$$j = 1,\ 2,\ \ldots,\ 1+[(n+1-i)/2]$$

The $\beta$ parameters are calculated via a similar algorithm involving the entries from the $\alpha$-table and the numerator coefficients. The first two rows of the $\beta$-table are obtained from the numerator coefficients giving a table of the form:

$\underline{\beta\text{-table}}$

$$\begin{array}{cccccc} B_{11} & B_{12} & \cdots & \cdots & B_{1,[(n+1)/2]} \\ B_{21} & B_{22} & \cdots & \cdots & B_{2,[(n+2)/2]} \\ B_{31} & B_{32} & \cdots & B_{3,[n/2]} & \\ \vdots & \vdots & & \vdots & \\ B_{n,1} & & & & \end{array}$$

with $B_{1,j} = b_{n-(2j-1)}$ and $B_{2,j} = b_{n-2(j-1)}$

$$j = 1,\ 2,\ \ldots,\ [(n+1)/2]$$

the remaining terms being given by the algorithm

$$B_{i,j} = B_{i-2,j+1} - \beta_{i-2} A_{i-1,j+1},\quad \beta_{i-2} = \frac{B_{i-2,1}}{A_{i-1,1}}$$

$$i = 3,\ 4,\ \ldots,\ n$$

$$j = 1,\ 2,\ \ldots,\ 1+[(n+1-i)/2]$$

Once the $\alpha$- and $\beta$-tables have been constructed, a $k$th order 'Routh convergent' can be formed by truncating the last ($n$ - $k$) terms of the continued fraction expansion given in (3.16). This may then be expressed in transfer function form as

$$R(s) = \frac{d_{k-1}s^{k-1} + \cdots + d_1 s + d_0}{e_k s^k + e_{k-1}s^{k-1} + \cdots + e_1 s + e_0} = \frac{N_k(s)}{D_k(s)}$$

by inverting the continued-fraction form.

Alternatively, both the reduced numerator and the denominator polynomials may be calculated recursively using the $\alpha_i$ and $\beta_i$ from the tables. For $k = 1, 2, \ldots, n-1$, $N_k(s)$ can be computed using the formula

$$N_k(s) = \alpha_k s N_{k-1}(s) + N_{k-2}(s) + \beta_k$$

where $N_{-1}(s) = 0$ and $N_0(s) = 0$ while for $D_k(s)$ we use the formula

$$D_k(s) = \alpha_k s D_{k-1}(s) + D_{k-2}(s)$$

with $D_{-1}(s) = 0$ and $D_0(s) = 1$. With the Routh convergent calculated in this fashion, the approximation produced tends to reflect the transient (high-frequency) behaviour of the full system because the division used to generate the continued-fraction expansion is from highest powers. However, since the steady-state (low-frequency) behaviour of $G(s)$ is usually of importance then Hutton and Friedland (1975) propose a "reciprocal" transformation $G(s) \rightarrow \hat{G}(s) = \frac{1}{s} G\left(\frac{1}{s}\right)$ before the expansion and truncation procedure and an inverse reciprocal transformation $\hat{R}(s) \rightarrow R(s) = \frac{1}{s}\hat{R}\left(\frac{1}{s}\right)$ for the reduced order model so that the steady-state behaviour is modelled accurately. Further, a useful measure of how good the Routh approximation is can be obtained from the $\alpha$ and $\beta$ parameters where,

$$I_n = \sum_{i=1}^{n} \frac{\beta_i^2}{2\alpha_i} = \int_0^\infty g^2(t)dt$$

is the "impulse response energy" of the system, where $g(t)$ is the system's impulse response. Hence, for the reduced system,

$$I_k = \sum_{i=1}^{k} \frac{\beta_i^2}{2\alpha_i}$$

and $I_k / I_n$ can be used as a norm for approximation purposes.

Also of interest is the discrete-time analogue of the Routh method developed by Shamash and Feinmesser (1978). They propose a modified Routh array for the calculation of the $\alpha$- and $\beta$-tables, which is equivalent to the algorithm for generating the table for the modified Jury test (Chen and Chan 1985) for the stability of discrete-time systems.

It is the stability preservation property of this method which has given rise to its enduring popularity, despite the continued-fraction and Padé approximation techniques which, if stable, tend to give better approximations of the full system.

**The Stability Equation Method**

Another major stability preserving approach is the "stability equation" method of Chen *et al* (1979). As well as the stability preservation property, the method has the advantage of retaining the first two time moments of the full system, thus matching steady-state responses between the full and reduced models for impulse, step and ramp inputs. Variants on the method have also appeared in the literature (e.g., Chen *et al* 1980, Parthasarthy and Jayasimha 1982, Therapos 1983).

When reducing an *n*th-order system to an ($n$ - 1)th-order model the stability equation method consists of splitting both the numerator and denominator polynomials of

$$G(s) = \frac{b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \cdots + b_0}{a_n s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \cdots + a_0}$$

into their 'alternant' polynomials defined as the odd and even degree terms respectively of those functions (Chen *et al* 1979). Consider the denominator polynomial

$$D(s) = a_n s^n + a_{n-1}s^{n-1} + \cdots + a_0$$

split into the alternant form,

$$D(s) = (a_n s^n + a_{n-2}s^{n-2} + \cdots + a_0) + s(a_{n-1}s^{n-2} + a_{n-3}s^{n-4} + \cdots + a_1)$$

where $n$ is taken as even without loss of generality. This may be written in factorised form as

$$D(s) = a_0\left(\frac{s^2}{z_1^2}+1\right)\left(\frac{s^2}{z_2^2}+1\right)\cdots\left(\frac{s^2}{z_{n/2}^2}+1\right) + a_1 s\left(\frac{s^2}{p_1^2}+1\right)\left(\frac{s^2}{p_2^2}+1\right)\cdots\left(\frac{s^2}{p_{(n-2)/2}^2}+1\right)$$

where the $z_i$ ($i = 1, 2, \ldots, n/2$) and $p_j$ ($j = 1, 2, \ldots, (n-2)/2$) satisfy the stability criterion (Chen *et al* 1979, Lucas 1992)

$$0 < |z_1| < |p_1| < |z_2| < |p_2| < \cdots < |p_{(n-2)/2}| < |z_{n/2}|.$$

The reduced model denominator polynomial of degree ($n$ - 1) is then generated by discarding the factor $\left[(s^2 / z_{n/2}^2) + 1\right]$. Since the stability criterion is still satisfied by the remaining factors a stable reduced model is guaranteed. This alternant factor process is then applied to the numerator $N(s)$ to obtain a reduced degree numerator. It should be noted that this method yields best results on minimum-phase systems (all zeros in the left half plane).

To apply the method to discrete-time systems, Chen *et al* (1979) make use of the bilinear transformation

$$z = \frac{1+w}{1-w}$$

on the discrete-time system $z$-transfer function. This transformation maps the inside of the unit circle $|z| = 1$ to the left half-plane $\mathrm{Re}(w) < 0$. This effectively changes the model reduction problem back to being one involving the reduction of a high order continuous-time system as above. Once a reduced transfer function is derived using the stability equation method the inverse transformation

$$w = \frac{z-1}{z+1}$$

is applied, transferring the system back to the $z$-domain thus obtaining the desired reduced discrete-time system.

The main disadvantage of this technique compared to the Routh method is clear from the above description. The roots/factors of the alternant polynomials must be determined, and for full systems of order greater than five this means the factorisation of cubic or higher degree polynomials. However, this computationally unattractive feature of the stability equation method has recently been overcome to some extent by the introduction of a recursive tabular approach due to Lucas (1992).

**Illustrative Example**

To illustrate and compare the stability preservation methods described in the preceding sections, consider the sixth-order transfer function

$$G(s) = \frac{s^5 + 17.5s^4 + 111s^3 + 314.5s^2 + 388s + 168}{s^6 + 15s^5 + 93s^4 + 307s^3 + 562s^2 + 562s + 260}$$

(with roots at -1, -1.5, -4, -4, -7 and poles at $-1\pm i,\ -2,\ -3\pm 2i,\ -5$) which is reduced to a third-order and a second-order model respectively, using these three methods.

Firstly, the Pole Retention method is applied. By retaining the three poles closest to the imaginary axis at $-1\pm i$ and $-2$, and then matching the first three time moments to obtain the numerator coefficients, the reduced model is given by

$$R_3^P(s)=\frac{1.58329s^2+4.2594s+2.58461}{s^3+4s^2+6s+4}$$

which has the relative impulse and step integral square errors

$$I_{rel}=3.5257\% \quad \text{and} \quad J_{rel}=1.0932\%$$

When only the conjugate pair of poles closest to the imaginary axis and two time moments are retained the result is the second-order reduced model

$$R_2^P(s)=\frac{1.48354s+1.2923}{s^2+2s+2}$$

giving

$$I_{rel}=2.665\% \quad \text{and} \quad J_{rel}=1.0543\%$$

This demonstrates clearly that good approximation can result using this method even when the original transfer function, $G(s)$, has two pairs of complex poles. The lower I.S.E.s for the second-order model may be explained by the fact that that the zero at $-1.766$ in the third order model is "close" to cancelling with the pole at -2.

Secondly, applying the Routh Approximation method, as described with reciprocal transformation, gives the following reduced models with relative integral square errors quoted for reference:

$$R_3^R(s)=\frac{1.45995s^2+2.0225s+0.87566}{s^3+2.65177s^2+2.9295s+1.3552}$$

with

$$I_{rel} = 3.5156\%, \quad J_{rel} = 2.9872\%$$

and

$$R_2^R(s) = \frac{0.92388s + 0.4}{s^2 + 1.33819s + 0.61905}$$

with

$$I_{rel} = 17.8985\%, \quad J_{rel} = 57.5058\%.$$

It is observed that the Routh method matches the first $k$ time moments of the full and reduced systems.

Finally, using the Stability Equation method the following reduced models are obtained:

$$R_3^S(s) = \frac{304.9s^2 + 388s + 168}{276.5s^3 + 515.3s^2 + 562s + 260}$$

giving

$$I_{rel} = 7.9819\%, \quad J_{rel} = 18.1661\%$$

and

$$R_2^S(s) = \frac{388s + 168}{515.3s^2 + 562s + 260}$$

with

$$I_{rel} = 27.1111\%, \quad J_{rel} = 92.8313\%.$$

Figures 3.1 and 3.3 show the step and impulse responses respectively of the full system $G(s)$ and the third order reduced models produced by the stability-preserving methods, $R_3^P(s)$, $R_3^R(s)$ and $R_3^S(s)$. A similar comparison of the second order models, $R_2^P(s)$, $R_2^R(s)$ and $R_2^S(s)$, is given in figures 3.2 and 3.4.

**Fig 3.1 Third Order Models**

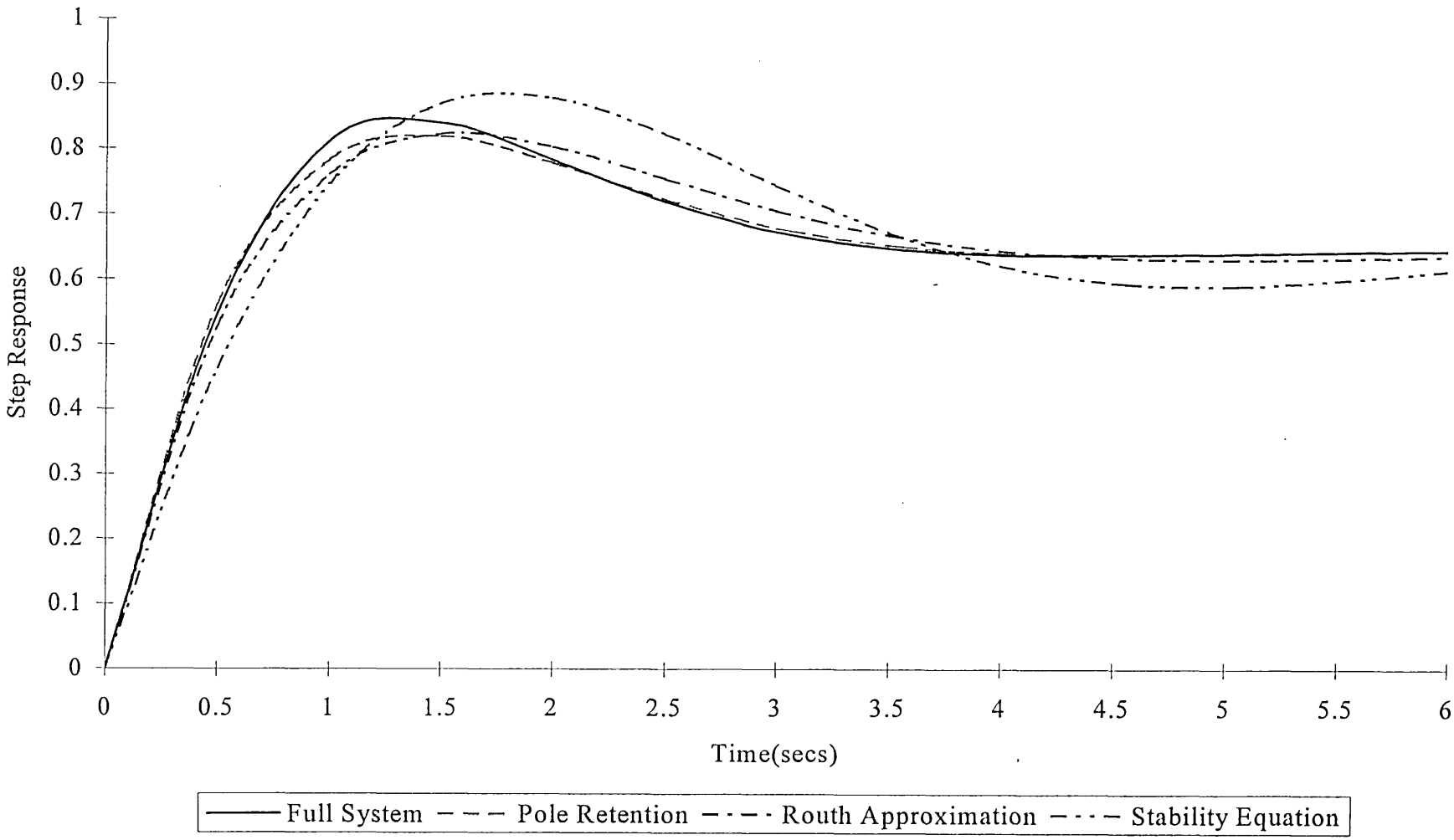


**Fig 3.2 Second Order Models**

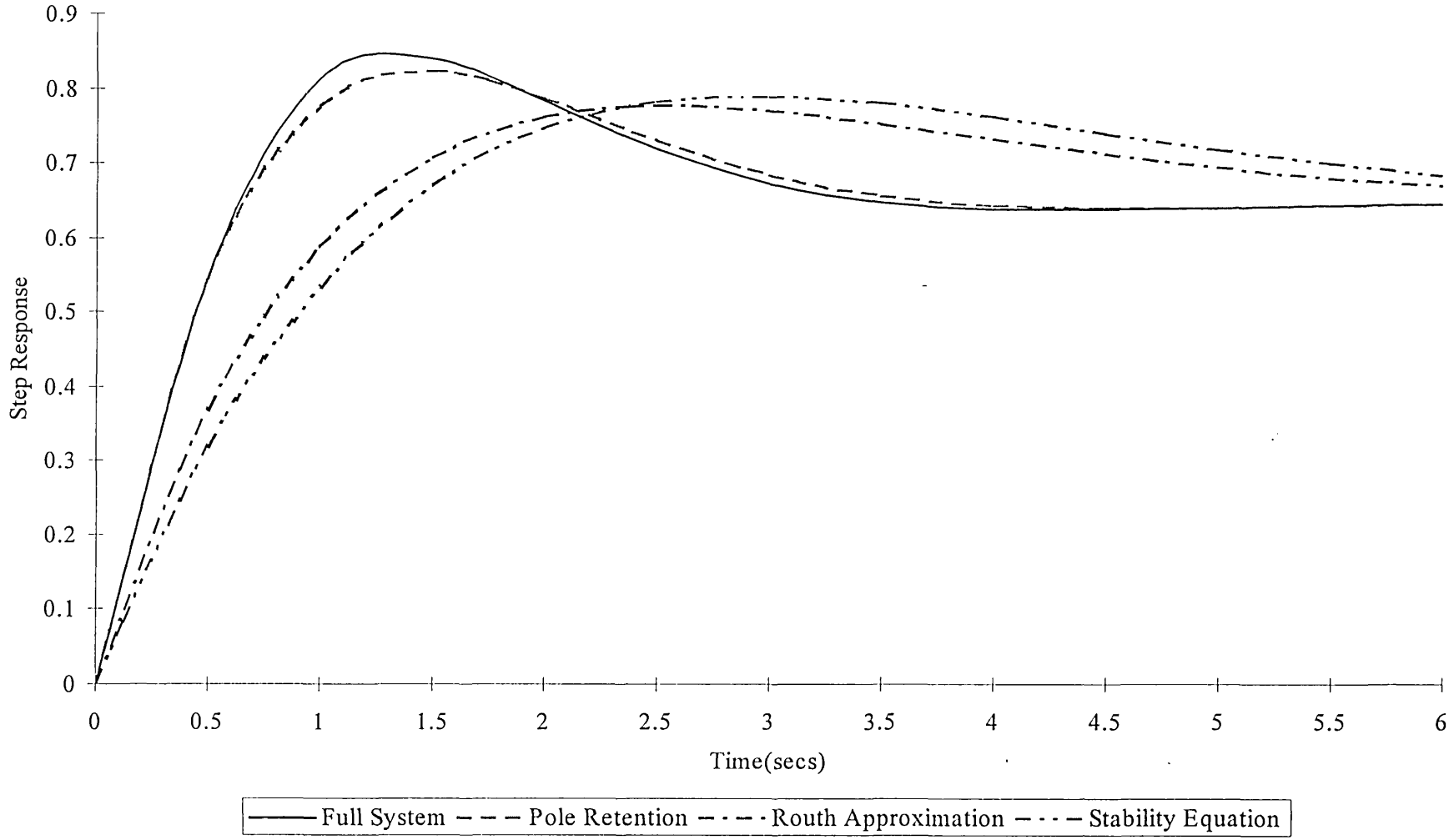


**Fig 3.3 Third Order Models**

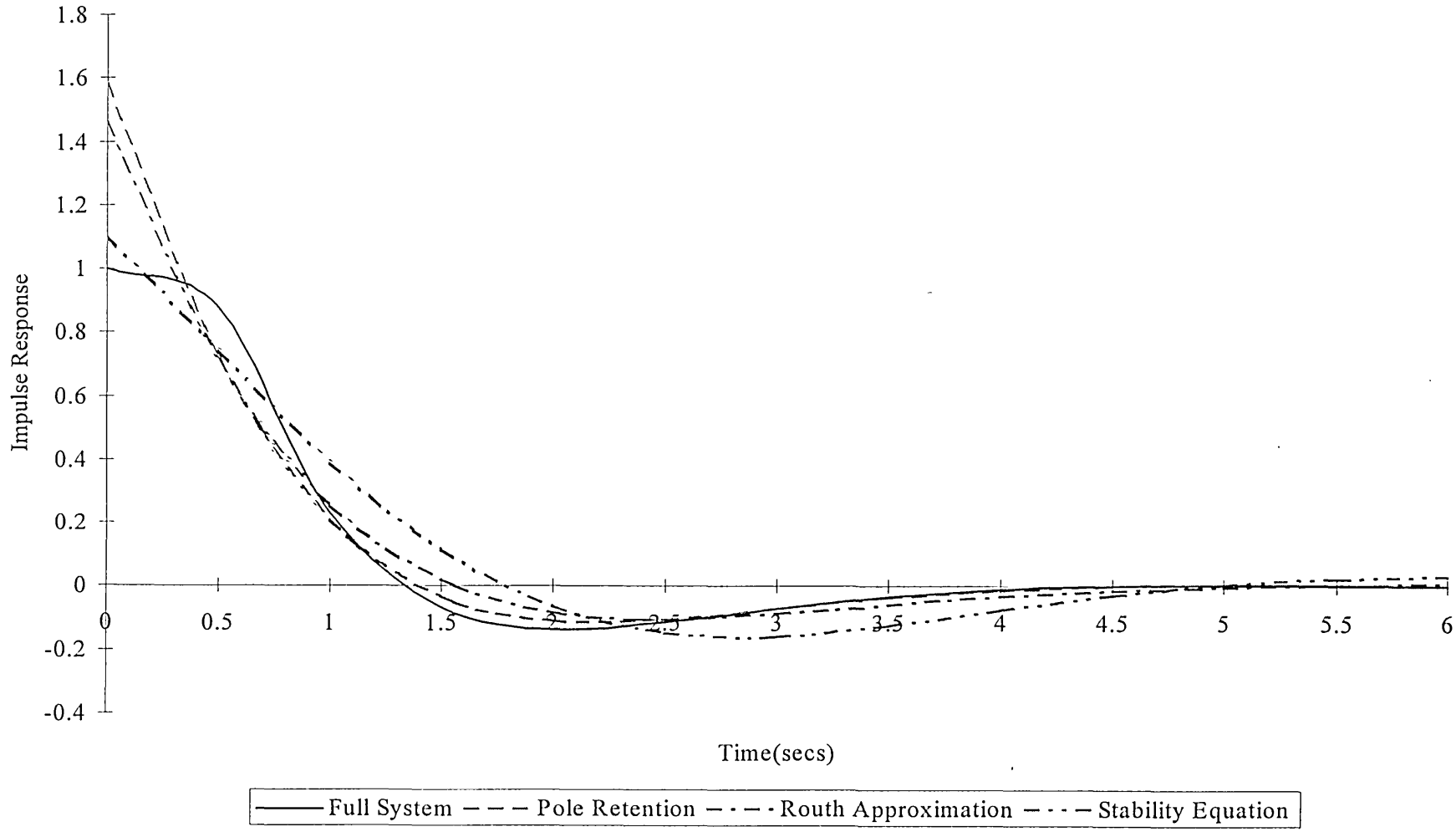
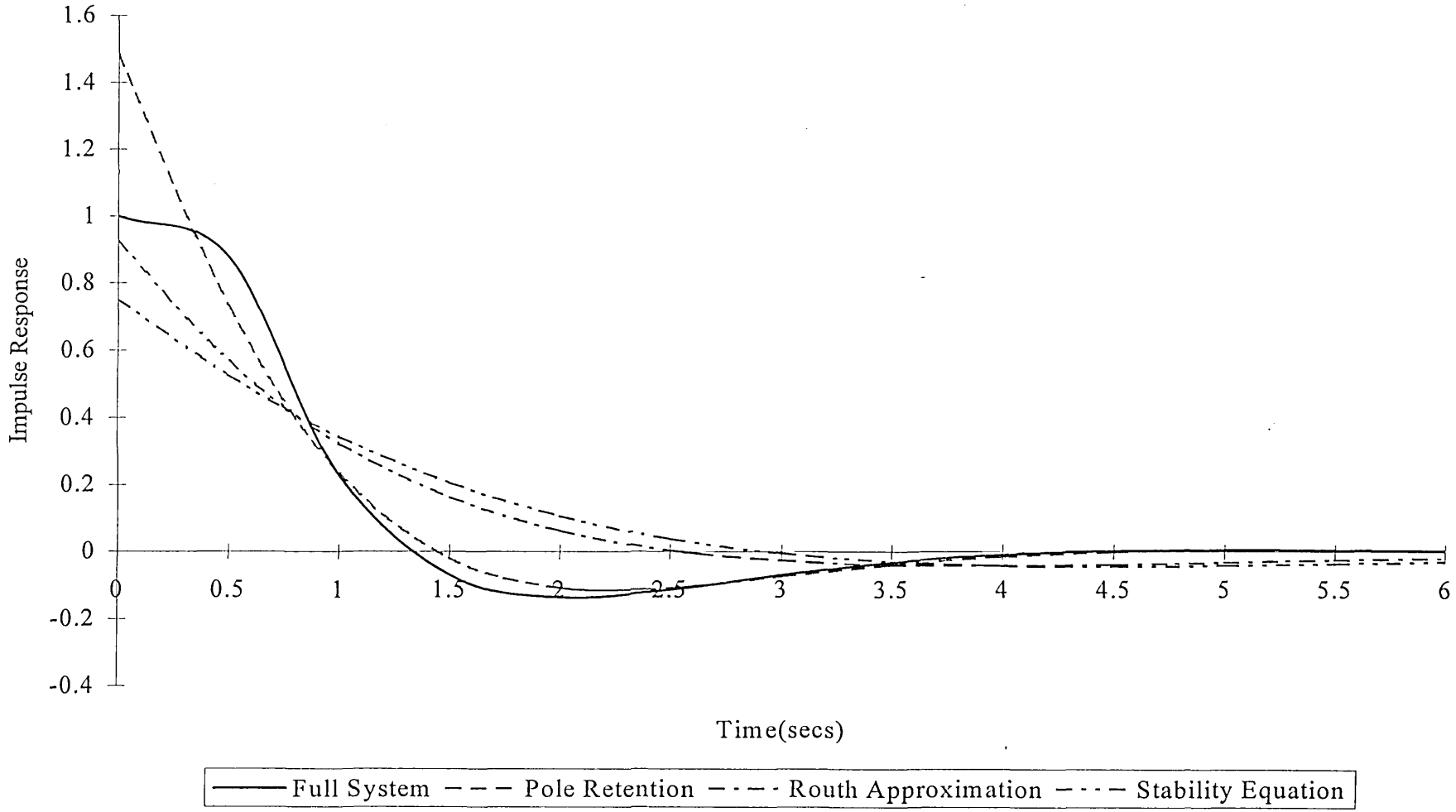


**Fig 3.4 Second Order Models**



# CHAPTER 4

# LEAST-SQUARES PADÉ APPROXIMATION

## 4.1 Introduction

In chapter 3 we considered two divergent approaches to improving the robustness of the exact Padé method of model reduction. The first of these approaches is the extension of the classical method of moment-matching to matching system parameters for values of $s$ other than zero giving the multi-point Padé method. The second is the development of entirely different methods which give robustness by way of guaranteed stability of the reduced model given a stable full system.

This chapter will explore a third alternative for improving the robustness of the exact Padé method in the form of "least-squares" (LS) Padé approximation. The general idea underlying this approach is to use more than the usual $2k$ items of system information to calculate a $k$th order model in an attempt to overcome the problems caused by the generation of unsuitable reduced models or singularity of $A$ in (3.10). As LS Padé approximation is the main focus of this thesis it is essential to provide fairly detailed explanations of the work carried out so far in this area of research.

The classical or 'exact' Padé method derives $2k$ numerator and denominator coefficients of a $k$th order reduced model from a system of $2k$ Padé equations such as (3.8). Essentially, the idea of an LS Padé approximation involves deriving the $2k$ numerator and denominator coefficients from a system of $2k + 1$ or more Padé equations. The over-determined set of equations is then solved in a least-squares sense.

LS Padé methods were first proposed for discrete-time systems by Yahagi (1980) (although he did not refer to it as such) and for continuous-time systems by

Shoji *et al* (1985). This work on continuous-time systems was further developed in the literature by Lucas and Beat (1990), Lucas and Munro (1991) and Aguirre (1992). In contrast, the application of LS Padé methods to discrete-time systems appears to have been pursued only by Lalonde (1992a) and Lalonde *et al* (1992b).

A consideration of these developments proves to be revealing and the following outline is adopted for the chapter. Section 4.2 prefaces the more detailed explanations with some general comments about "least-squares" approximation. In section 4.3, the LS Padé method as it was originally applied to moment matching in the reduction of continuous-time systems (Shoji *et al* 1985, Lucas and Beat 1990) is examined. In section 4.4, the development of the method for the production of biased reduced models (Lucas and Munro 1991, Aguirre 1992) using both time moment and Markov parameter information is considered. Section 4.5 outlines the application of LS approximation to the calculation of the numerator of the reduced order model. Finally, in section 4.6, an account is given of the main features of application to the reduction of discrete-time systems using Markov parameters only (Yahagi 1980, Lalonde *et al* 1992).

## 4.2 Least-squares Approximation

It is well known that the optimization technique referred to as the "least-squares method" may be interpreted mathematically as the approximate solution of an overdetermined set of linear equations of the form

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1,n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2,n}x_n = b_2$$

$$\vdots \qquad\qquad \vdots$$

$$a_{m,1}x_1 + a_{m,2}x_2 + \ldots + a_{nm} = b_m$$

where $m > n$. This set of equations may be written in the matrix-vector form

$$A\mathbf{x} = \mathbf{b}$$

where the elements of the matrix $A$ are the left-hand side coefficients $a_{ij}$ with $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$ and the vector $\mathbf{b}$ contains $b_i (i = 1, 2, \ldots, m)$. Premultiplying both sides of this equation by the generalised inverse $\left(A^T A\right)^{-1} A^T$ gives the solution of the overdetermined linear system in a least-squares sense and minimises the Euclidean norm of the vector $A\mathbf{x} - \mathbf{b}$.

Thus outlined, the least-squares method is an extremely versatile technique provided care is exercised concerning the possible "ill-conditioned" nature of the set of equations involved. If this is the case, perturbation of the values of the coefficients $a_{ij}$ will lead to large changes in the approximation for $\mathbf{x}$ produced. One way of checking for ill-conditioning is to check the "normalized" determinant (Conte, 1965) of the matrix $A^T A$ given by $\det(A^T A)$ divided by the product $\rho_1 \rho_2 \ldots \rho_n$ where

$$\rho_i = \sqrt{l_{i,1}^2 + l_{i,2}^2 + \ldots + l_{i,n}^2}$$

the $l_{ij} (j = 1, 2, \ldots, n)$ being the elements of the $i$th row of $A^T A$. The matrix is said to be ill-conditioned if the normalized determinant is "small" compared to unity. In what follows it is assumed that such checks for possibe numerical difficulties are applied. Also, it should be noted that, to achieve the highest level of numerical accuracy throughout, double-precision arithmetic is used in all the software written or adapted specially for the project and, where examples have been calculated using the mathematical package DERIVE, exact arithmetic is used with approximation taking place only in the final step.

## 4.3 Least-Squares Moment Matching

It is well-known (section 3.2) that the exact Padé method involves matching the first $2k$ time moments of $G(s)$ to those of the reduced model $R(s)$. This approach leads to reduced order models which often give good steady-state approximations to the full system response, but which may lead to unstable reduced order models being derived from stable systems. In an attempt to ameliorate this situation Shoji *et al* (1985) first proposed extending the number of time moments included in the calculation until a suitable/stable reduced model is produced.

Consider the matrix-vector equation obtained from the last $k$ equations of (3.8) which contain the denominator polynomial coefficients $e_i$, $(i = 0, 1, 2, \ldots, k-1)$ of $R(s)$ defined in (3.4), i.e.

$$\begin{bmatrix} -c_k & -c_{k-1} & \cdots & -c_1 \\ -c_{k+1} & -c_k & \cdots & -c_2 \\ \vdots & \vdots & \ddots & \vdots \\ -c_{2k-1} & -c_{2k-2} & \cdots & -c_k \end{bmatrix} \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{k-1} \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{k-1} \end{bmatrix} \tag{4.1}$$

The basic proposal was that if the solution of (4.1) gives a denominator polynomial for $R(s)$ such that the reduced model is unsuitable then a further equation is added to the set represented by (4.1). In other words, the next time moment from the full system is assumed to be matched to the reduced model giving

$$H\mathbf{e} = \mathbf{c}$$

$$\text{where } H = \begin{bmatrix} -c_k & -c_{k-1} & \cdots & -c_1 \\ \vdots & \vdots & & \vdots \\ -c_{2k-1} & -c_{2k-2} & \cdots & -c_k \\ -c_{2k} & -c_{2k-1} & \cdots & -c_{k+1} \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{k-1} \end{bmatrix} \text{ and } \mathbf{c} = \begin{bmatrix} c_0 \\ \vdots \\ c_{k-1} \\ c_k \end{bmatrix} \tag{4.2}$$

which, because $H$ is now no longer a square matrix, must be solved in the least-squares sense making use of the generalized inverse of $H$, so that

$$\mathbf{e} = \left(H^T H\right)^{-1} H^T \mathbf{c} \qquad (4.3)$$

and the reduced numerator polynomial is obtained by exact moment matching between the full and reduced models. This approximation may still yield an unsuitable model and, in this case, $H$ and $\mathbf{c}$ in (4.2) are extended by a further row, which assumes the matching of the next time moment from $G(s)$ in the LS calculation. This process is carried on until a suitable reduced model results.

The effectiveness of this suggestion can be readily appreciated by looking again at the sixth-order example used for illustration in section 3.3, i.e.

$$G(s) = \frac{s^5 + 17.5s^4 + 111s^3 + 314.5s^2 + 388s + 168}{s^6 + 15s^5 + 93s^4 + 307s^3 + 562s^2 + 562s + 260}$$

When the exact Padé method is applied to this system the third-order reduced model produced is unstable and the LS method of Shoji *et al* (1985) may be applied. The result of using one further time moment is striking in this case giving

$$R(s) = \frac{1.42832s^2 + 1.88967s + 0.505489}{s^3 + 2.27159s^2 + 2.80872s + 0.782306}$$

with impulse and step relative integral square errors of

$$I_{rel} = 1.9678\% \quad \text{and} \quad J_{rel} = 0.5737\%$$

respectively. The graphs of the impulse and step responses of the full and reduced models are compared in figures 4.1 and 4.2 respectively.

Although the performance of the technique in the above example is good, the claims of Shoji *et al* (1995), that the method would not only overcome possible singularity problems, but also result in a stable reduced model if enough time moments are used in extending the number of rows in $H$ and $\mathbf{c}$, prove to be excessive. Lucas and Beat (1990) demonstrate that this LS Padé method is in fact very sensitive to the number of extra time moments used and, indeed, to the pole distribution of the full system. In

**Fig 4.1 Least-squares Moment Matching**

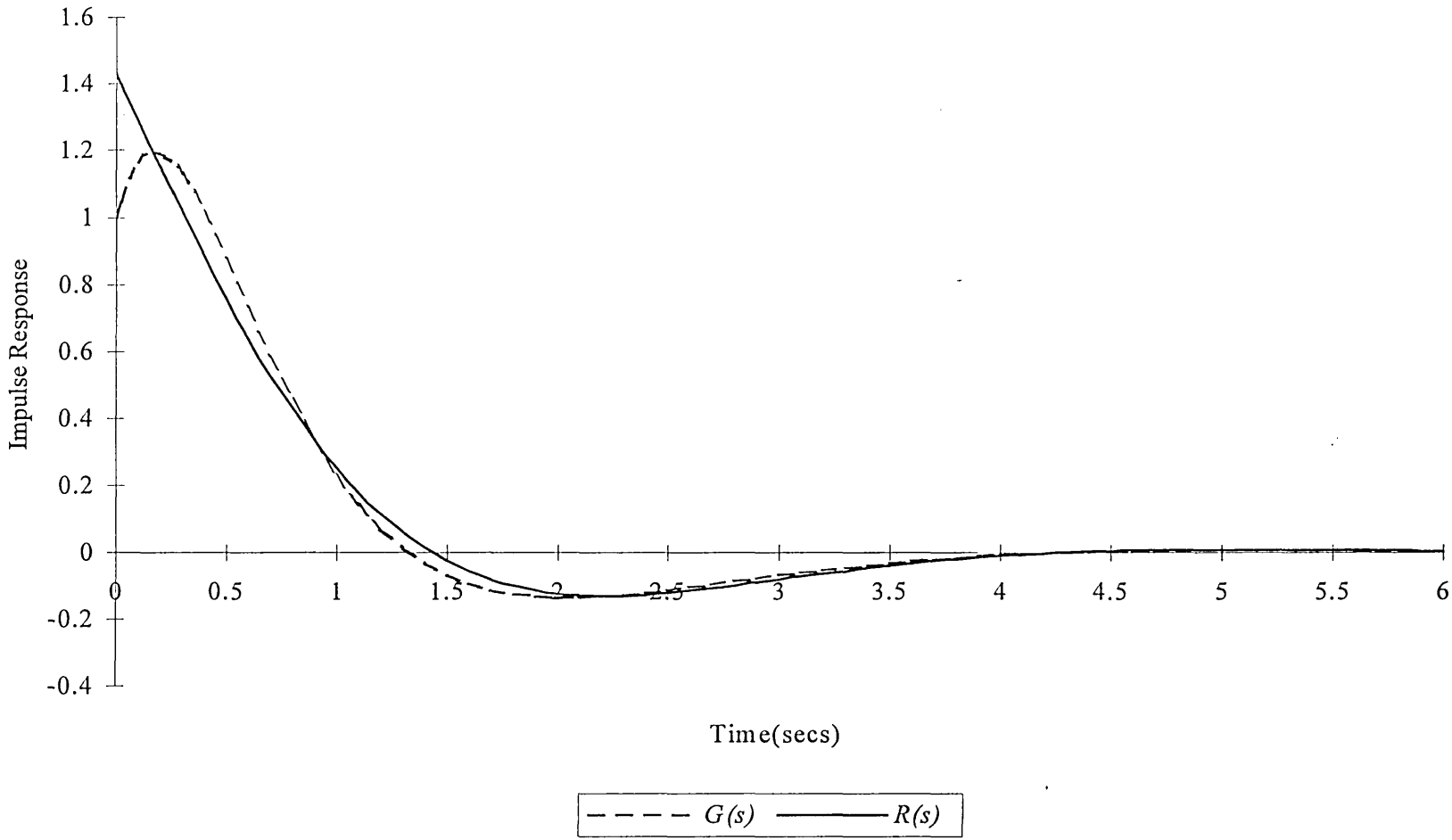
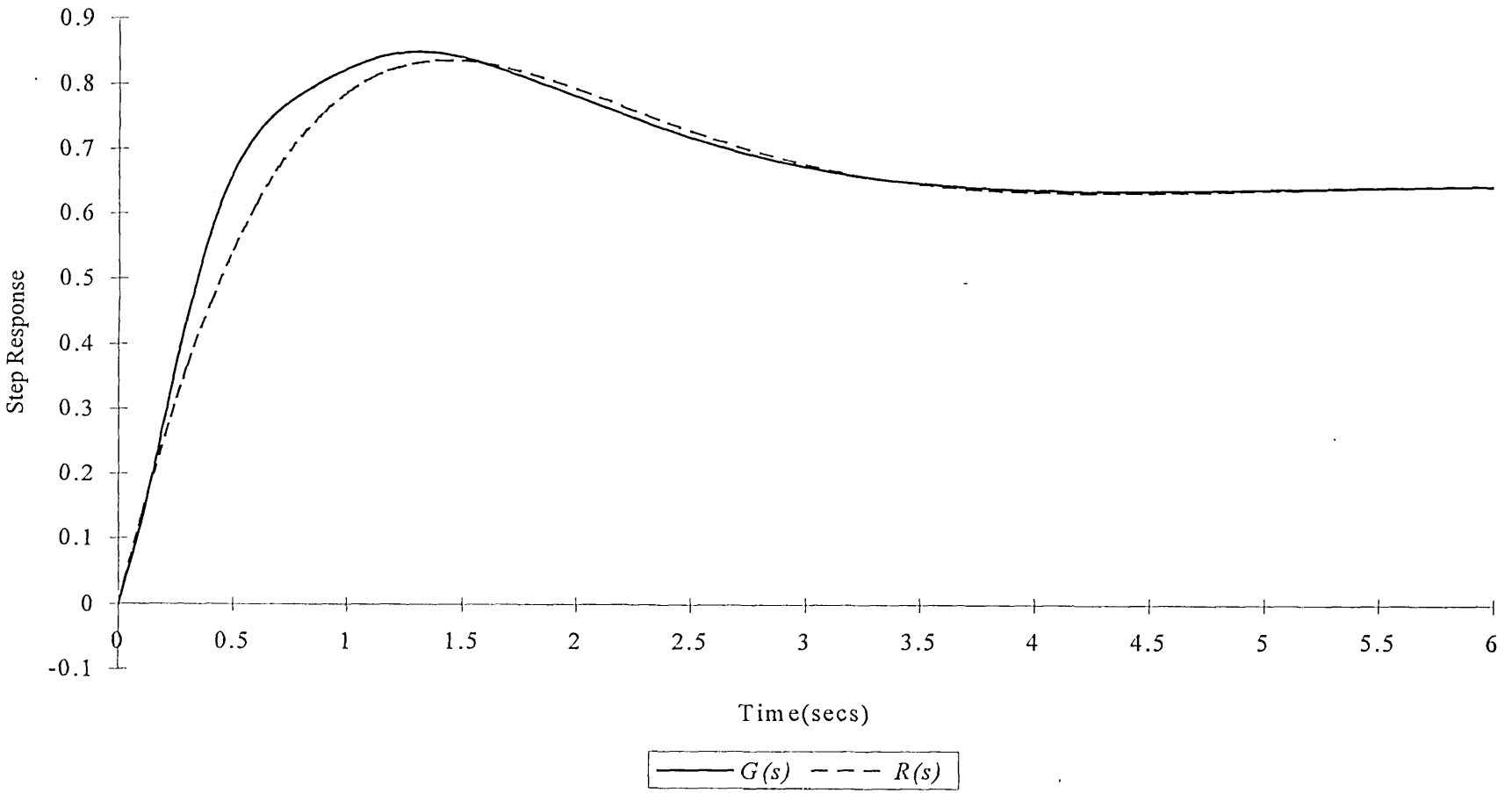


**Fig 4.2 Least-squares Moment Matching**



particular, they point to the comparison of the expansion of $G(s)$ about $s = 0$ with its partial fraction expansion

$$G(s) = \sum_{m=1}^{n} \frac{r_m}{(s + p_m)} \tag{4.4}$$

in terms of the poles $-p_m$ and the corresponding residues $r_m$. They note that the Taylor series expansion of (4.4) gives the time moments as

$$c_i = (-1)^i \sum_{m=1}^{n} \frac{r_m}{p_m^{i+1}}$$

This shows immediately the sensitivity of the method to the pole distribution because $c_i \to \infty$ as $i \to \infty$ when the full system has any pole $-p_m$ such that $|p_m| < 1$. Having also observed that, when the smallest magnitude pole *is* greater than unity the process might still give an unstable reduced denominator, Lucas and Beat (1990) propose use of the linear shift $s \to s+a$ to ensure that the smallest magnitude pole is close to unity. Thus, applying the method of LS moment matching to the transformed system $G(s+a)$, with smallest magnitude pole near or equal to unity, a reduced denominator is formed from **e** as in (4.3) and the inverse shift $s \to s-a$ applied before matching the time moments of $G(s)$ for the reduced numerator calculation. An indication of the kind of improvement achieved using this modification of LS moment matching is also given by the authors.

## 4.4 Generalised Least-Squares Method

Lucas and Munro (1991) extend the LS moment matching proposed by Shoji *et al* (1985) to include Markov parameters in the process to give better matching of the transient responses of the full and reduced models. The exact Padé method for producing biased two point approximations may be developed easily by considering the case where a $k$th order reduced model is derived by matching $k + t$ time moments

and $k$ - $t$ Markov parameters $(0 \le t \le k)$ of the full system. In this case, the numerator and denominator coefficients of the reduced model

$$R(s) = \frac{d_{k-1}s^{k-1} + \cdots + d_1 s + d_0}{s^k + e_{k-1}s^{k-1} + \cdots + e_1 s + e_0}$$

are derived from the following sets of equations

$$\begin{aligned}
d_0 &= e_0 c_0 \\
d_1 &= e_0 c_1 + e_1 c_0 \\
&\vdots \\
d_{k-1} &= e_0 c_{k-1} + e_1 c_{k-2} + \cdots + e_{k-1} c_0 \\
0 &= e_0 c_k + e_1 c_{k-1} + \cdots + e_{k-1} c_1 + c_0 \\
&\vdots \\
0 &= e_0 c_{k+t-1} + e_1 c_{k+t-2} + \cdots + e_{t-1} c_t + c_{t-1}
\end{aligned} \tag{4.5}$$

being the Padé equations for the matched time moments, $c_i$ $(i = 0,\ 1,\ \ldots\ ,\ k+t-1)$, and

$$\begin{aligned}
d_{k-1} &= m_1 \\
&\vdots \\
d_t &= m_{k-t} + m_{k-t-1} e_{k-1} + \cdots + m_2 e_{t+2} + m_1 e_{t+1}
\end{aligned} \tag{4.6}$$

for the matched Markov parameters, $m_j$ $(j = 1,\ 2,\ \ldots\ ,\ k-t)$. Substituting from (4.6) for the $d_i$ $(i = t, t+1, \ldots, k-1)$ in (4.5) gives the following matrix-vector form for the two point exact Padé method

$$\begin{bmatrix} c_{k+t-1} & c_{k+t-2} & \cdots & \cdots & \cdots & \cdots & c_t \\ c_{k+t-2} & c_{k+t-3} & \cdots & \cdots & \cdots & c_t & c_{t-1} \\ \vdots & \vdots & & & & \vdots & \vdots \\ c_{k-1} & c_{k-2} & \cdots & \cdots & \cdots & c_1 & c_0 \\ c_{k-2} & c_{k-3} & \cdots & \cdots & \cdots & c_0 & -m_1 \\ c_{k-3} & c_{k-4} & \cdots & \cdots & c_0 & -m_1 & -m_2 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ c_t & c_{t-1} & \cdots & c_0 & -m_1 & \cdots & -m_{k-t-1} \end{bmatrix} \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{k-1} \end{bmatrix} = \begin{bmatrix} -c_{t-1} \\ -c_{t-2} \\ \vdots \\ -c_0 \\ m_1 \\ m_2 \\ m_3 \\ \vdots \\ m_{k-t} \end{bmatrix} \quad (4.7)$$

which is a generalisation of (4.1) for moment matching only. The numerator coefficients are then calculated using some suitable combination of time moment and Markov parameter matching.

This generalised two point LS method involves extending either (4.5) or (4.6) in the event of the denominator given by **e** in (4.7) being unstable or the system parameter matrix, $H$, being singular. Once the appropriate row has been added to the top or bottom of $H$ and **c** respectively, the LS solution of the resulting non-square system of equations is used to calculate **e**. Care should be exercised in applying this method when considering the number of Markov parameters to use in the process, since their sequence grows unbounded whenever (as is often the case) there are any system poles greater than unity in magnitude. However, Lucas and Munro (1991) illustrate, by example, how the method may be used to improve upon that of Lucas and Beat (1990) mentioned previously.

All the least-squares Padé methods considered so far (Shoji *et al* 1985, Lucas and Beat 1990, and Lucas and Munro 1991) shall be referred to as *partial* LS methods. This terminology is used because only the denominator coefficients are approximated in the LS sense before the numerator coefficients are derived by exact moment matching or, in the case of Lucas and Munro, matching a mixture of time

moments and Markov parameters between the full and reduced models. In contrast, an account is now given of the LS Padé method proposed by Aguirre (1992). Here the author includes not only time moments and Markov parameters in the least-squares matching process but also information on retained dominant poles. This will be left out for presentational simplicity. As in the method of Lucas and Munro (1991), the starting point for Aguirre's method is the information in the equation sets (4.5) and (4.6) with extra time moment and/or Markov parameter equations being added to them if a suitable reduced order model is not forthcoming from the exact Padé method. To find a $k$th order reduced model using the first $2k + t$ time moments and first $r$ Markov parameters ($t > 0$ and $r < k$) the following set of linear equations results

$$A\mathbf{x} = \mathbf{b} \tag{4.8}$$

where

$$A = \begin{bmatrix} 0 & 0 & \cdots & 0 & -c_{k+t} & -c_{k+t+1} & \cdots & -c_{2k+t-1} \\ 0 & 0 & \cdots & 0 & -c_{k+t-1} & -c_{k+t} & \cdots & -c_{2k+t-2} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & -c_1 & -c_2 & \cdots & -c_k \\ 1 & 0 & \cdots & 0 & -c_0 & -c_1 & \cdots & -c_{k-1} \\ 0 & 1 & \cdots & 0 & 0 & -c_0 & \cdots & -c_{k-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & -c_0 \\ 1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \cdots & 0 & -m_1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & -m_{r-1} & -m_{r-2} & \cdots & 0 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} d_{k-1} \\ d_{k-2} \\ \vdots \\ d_0 \\ e_{k-1} \\ e_{k-2} \\ \vdots \\ e_0 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} c_{k+l-1} \\ c_{k+l-2} \\ \vdots \\ c_0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ m_1 \\ m_2 \\ \vdots \\ m_r \end{bmatrix}$$

Equations (4.8) may now be solved in the LS sense using the generalised inverse to obtain

$$\mathbf{x} = \left(A^T A\right)^{-1} A^T \mathbf{b} \tag{4.9}$$

which calculates both the numerator and denominator coefficients of the reduced model simultaneously. Therefore, it is appropriate to refer to this LS method as a *full* LS Padé method to distinguish it from the methods already described. The exact relationship between the full and partial approaches to LS Padé model reduction is not obvious, but example 4.1 clearly indicates that they do not always produce the same reduced order model when applied to a given full system.

As a follow up to this work, Aguirre (1994a) shows how stability preservation may be achieved by applying this LS Padé method to the simplification of squared magnitude functions, which can be given in the $s$-domain as $P(s^2) = G(s)G(-s)$. He further proposed a method which combines the LS Padé method with exact retention of certain poles and/or zeros of the full system, again to address the stability problem (Aguirre 1994b).

**Example 4.1**

Consider the fourth-order system given by the transfer function (Aguirre 1992)

$$G(s) = \frac{267s^3 + 527s^2 + 385s + 100}{s^4 + 4s^3 + 6s^2 + 4s + 1}$$

The third-order model by the exact Padé method is unstable and applying the partial LS method of Lucas and Munro (1991) for denominator derivation using 6 time moments and the first 3 Markov parameters, the numerator being found by exact moment matching, the model obtained is

$$R_P(s) = \frac{266.346s^2 + 345.083s + 101.469}{s^3 + 3.33582s^2 + 3.60303s + 1.01469}$$

However, when the full LS method of Aguirre (1992) is applied using the same number of time moments and Markov parameters the model obtained is

$$R_F^*(s) = \frac{267.595s^2 + 345.578s + 92.0478}{s^3 + 3.33228s^2 + 3.55884s + 0.896506}$$

Using Aguirre's idea of multiplying through by a constant factor to make the steady-state response of the reduced model match that of $G(s)$, the third-order model becomes

$$R_F(s) = \frac{260.2s^2 + 336.035s + 89.6506}{s^3 + 3.33228s^2 + 3.55884s + 0.896506}$$

From these results it is quite clear that neither form, $R_F^*(s)$ or $R_F(s)$, is identical to $R_P(s)$ obtained by the partial LS technique.

While this difference is consistent with the assumption that the full LS Padé method approximates all of the reduced model's coefficients such that there is a least-squares error between the full and reduced model's parameters, it still remains simply an assumption. Indeed, it is clear that the literature cited in this chapter consists of a number of apparently distinct LS model reduction techniques, with relatively little

work having been done on the analysis of exactly how the LS Padé method approximates the original high order system. It will be seen in chapter 5 that this and other widely-held assumptions have helped to obscure the underlying relationship between the various LS techniques.

## 4.5 Least-squares Approximation of the Numerator

A further extension of the LS Padé method is proposed by Aguirre (1995) where he considers the application of the technique to model reduction methods which determine the denominator prior to the numerator, e.g., pole retention. Aguirre notes that when using the $2k + t$ Padé equations

$$\begin{aligned}
d_0 &= e_0 c_0 \\
d_1 &= e_0 c_1 + e_1 c_0 \\
&\vdots \\
d_{k-1} &= e_0 c_{k-1} + e_1 c_{k-2} + \cdots + e_{k-1} c_0 \\
0 &= e_0 c_k + e_1 c_{k-1} + \cdots + e_{k-1} c_1 + c_0 \\
&\vdots \\
0 &= e_0 c_{2k+t-1} + e_1 c_{2k+t-2} + \cdots + e_{k+t-1} c_k + c_{k+t-1}
\end{aligned} \tag{4.10}$$

that the first $k + t$ equations cannot be used to perform an LS calculation of the numerator coefficients. This is because only $k$ of these equations involve the numerator coefficients. Indeed, at first sight it appears impossible to obtain an overdetermined set of equations involving the numerator coefficients. However, as Aguirre demonstrates, this difficulty may be overcome by redefining the time moments in terms of the reduced model's coefficients as follows

$$c_0 = \frac{d_0}{e_0}$$

$$c_i = \frac{d_i - \sum_{j=1}^{i} \alpha_j d_{i-j}}{e_0} \qquad i > 0$$

where

$$\alpha_0 = -1$$

$$\alpha_i = \frac{-\sum_{j=1}^{i} e_j \alpha_{i-j}}{e_0} \qquad i > 0$$

These definitions allow the first $k + t$ equations of (4.10) to be written in the form

$$A\mathbf{d} = e_0\mathbf{c} \tag{4.11}$$

where

$$\mathbf{d} = \begin{bmatrix} d_0 & d_1 & \cdots & d_{k-1} \end{bmatrix}^T$$

$$\mathbf{c} = \begin{bmatrix} c_0 & c_1 & \cdots & c_{k+t-1} \end{bmatrix}^T$$

and

$$A = \begin{bmatrix} 1 & \cdots & \cdots & \cdots & 0 \\ -\alpha_1 & 1 & \cdots & \cdots & 0 \\ -\alpha_2 & -\alpha_1 & \cdots & \cdots & 0 \\ \vdots & \vdots & & & \vdots \\ -\alpha_{k+t-1} & -\alpha_{k+t-2} & \cdots & \cdots & -\alpha_t \end{bmatrix}$$

The LS solution of (4.11)

$$\mathbf{d} = \left(A^T A\right)^{-1} A^T e_0 \mathbf{c}$$

is essentially the solution of the overdetermined system of $k + t$ equations

$$e_0 c_0 = d_0$$

$$e_0 c_1 = d_1 - \alpha_1 d_0$$

$$\vdots$$

$$e_0 c_{k-1} = d_{k-1} - \alpha_{k-1} d_0 - \ldots - \alpha_1 d_{k-2}$$

$$e_0 c_k = -\alpha_k d_0 - \alpha_{k-1} d_1 - \ldots - \alpha_1 d_{k-1}$$

$$\vdots$$

$$e_0 c_{k+t-1} = -\alpha_{k+t-1} d_0 - \alpha_{k+t-2} d_1 - \ldots - \alpha_t d_{k-1}$$

which all involve the numerator coefficients of a $k$th order reduced model for which the denominator coefficients are already determined. The estimated numerator is optimal in the sense that the Euclidean norm of the vector $A\mathbf{d} - e_0\mathbf{c}$ is minimised in terms of the error index $J$ where

$$J^2 = \sum_{i=0}^{k+t-1} \left( e_0 c_i - h^*_{i+1} \right)^2$$

where the $h^*_j$ are the product of the $j$th row of the matrix $A$ and the numerator coefficient vector $\mathbf{d}$. This not only provides for a wider choice regarding the way in which the LS Padé method may be applied in obtaining reduced order models but also provides the basis (see section 7.5) for an LS model reduction method applicable to multivariable systems (Aguirre and Mendes, 1995).

## 4.6 Discrete-Time Least-Squares Model Reduction

Just as the proposal of Shoji *et al* (1985) uses more than $2k$ time moments (for a $k$th order model) to overcome singularity and instability problems in the continuous-time case, so the proposal to match more than $2k$ Markov parameters is a logical extension of the exact Padé method in the discrete-time case (Yahagi 1980, Lalonde

*et al* 1992a). This is because of the property that the Markov parameters of a discrete-

time system are the same as the time response values to a pulse input. Yahagi (1980) seems to be the first in the literature to propose the method of an LS fit for obtaining a reduced order model as applied to a discrete-time system given by the $z$-transfer function

$$G(z) = \frac{b_n z^n + b_{n-1} z^{n-1} + \cdots + b_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_0}$$

However, his interest in the technique appears to have been mainly numerical and computational. Consequently, the LS technique developed is somewhat "hidden" and has remained uncited in the literature on LS methods.

Lalonde *et al* (1992a) have applied the method of LS system parameter matching to discrete-time systems using Markov parameters only. This approach is seen to exhibit a number of interesting features, and useful parallels with the application of LS moment matching to continuous-time systems are readily observed. Suppose that $G(z)$ is to be reduced to one of order $k$, given by

$$R(z) = \frac{d_k z^k + d_{k-1} z^{k-1} + \cdots + d_1 z + d_0}{z^k + e_{k-1} z^{k-1} + \cdots + e_1 z + e_0} \tag{4.12}$$

If $G(z)$ is expanded about $z = \infty$ then

$$G(z) = m_0 + m_1 z^{-1} + m_2 z^{-2} + \ldots \tag{4.13}$$

and $m_i$ ($i = 0, 1, 2, \ldots$) is the pulse response value at time $t = iT$, where $T$ is the sample time interval. It should be noted that the $m_i$ in (4.13) are easily obtained by division of the denominator of $G(z)$ into the numerator from highest powers of $z$.

This property makes it attractive to use the Padé approximation method with matching of Markov parameters, rather than time moments, to calculate reduced

order models (especially if impulse inputs are used). This would imply that, for exact matching of the first $2k + 1$ pulse response values of the system to those of the reduced model, the Padé equations to be solved would involve the first $2k + 1$ Markov parameters of $G(z)$. Applying the idea of extending the number of Markov parameters included in the calculation of the reduced $k$th order model, the following set of equations are formed in the same way as for LS moment matching in section 4.3 by equating (4.12) and (4.13)

$$\begin{aligned}
&d_k = m_0 \\
&d_{k-1} = m_0 e_{k-1} + m_1 \\
&\vdots \\
&d_0 = m_0 e_0 + m_1 e_1 + \cdots + m_{k-1} e_{k-1} + m_k \\
&0 = m_1 e_0 + m_2 e_1 + \cdots + m_k e_{k-1} + m_{k+1} \\
&0 = m_2 e_0 + m_3 e_1 + \cdots + m_{k+1} e_{k-1} + m_{k+2} \\
&\vdots \\
&0 = m_{k+r} e_0 + m_{k+r+1} e_1 + \cdots + m_{2k+r-1} e_{k-1} + m_{2k+r}
\end{aligned} \tag{4.14}$$

where $r$ is the number of extra Markov parameters used to extend the exact Padé method. As in the continuous-time case, these equations may be expressed in the matrix-vector form of

$$A\mathbf{x} = \mathbf{b} \tag{4.15}$$

where

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & -m_0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 & -m_{k-1} & -m_{k-2} & \cdots & -m_0 \\ 0 & 0 & \cdots & 0 & -m_k & -m_{k-1} & \cdots & -m_1 \\ 0 & 0 & \cdots & 0 & -m_{k+1} & -m_k & \cdots & -m_2 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & -m_{2k+r-1} & -m_{2k+r-2} & \cdots & -m_{k+r} \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} d_k \\ d_{k-1} \\ \vdots \\ d_0 \\ e_{k-1} \\ e_{k-2} \\ \vdots \\ e_0 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_k \\ m_{k+1} \\ m_{k+2} \\ \vdots \\ m_{2k+r} \end{bmatrix}$$

The reduced model's coefficients are then obtained by solving (4.15) by least-squares to give

$$\mathbf{x} = \left(A^T A\right)^{-1} A^T \mathbf{b}$$

It is seen that this method is a *full* LS Padé method like that of Aguirre (1992). A corresponding *partial* approach, as advocated by Yahagi (1980), would be to solve for the reduced model's denominator first by least-squares parameter matching, i.e. solve

$$H\mathbf{e} = \mathbf{q} \tag{4.16}$$

where

$$H = \begin{bmatrix} -m_k & -m_{k-1} & \cdots & -m_1 \\ -m_{k+1} & -m_k & \cdots & -m_2 \\ \vdots & \vdots & & \vdots \\ -m_{2k+r-1} & -m_{2k+r-2} & \cdots & -m_{k+r} \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} e_{k-1} \\ e_{k-2} \\ \vdots \\ e_0 \end{bmatrix} \text{ and } \mathbf{q} = \begin{bmatrix} m_{k+1} \\ m_{k+2} \\ \vdots \\ m_{2k+r} \end{bmatrix}$$

to give

$$\mathbf{e} = \left(H^T H\right)^{-1} H^T \mathbf{q}$$

followed by exact Markov parameter matching to obtain the numerator coefficients of the reduced model. It will be shown in chapter 6 that these two different approaches will in fact give the same reduced models and an interesting stability preservation property is proven.

# CHAPTER 5

# A FRAMEWORK FOR LEAST-SQUARES PADÉ METHODS

## 5.1 Introduction

In this chapter new results are presented concerning the methods described in chapter 4. An analysis of the LS approach to model reduction is offered with proofs given of some interesting properties which throw light upon the exact nature of LS Padé approximation as a model reduction method. This is achieved by placing all the apparently disparate LS Padé methods presented in the last chapter into a common framework.

This framework is described in its most general terms in section 5.5, which is based on a paper by Smith and Lucas (1996). Prior to the presentation of this framework, section 5.2 focuses on the analysis of LS moment-matching as a two-stage process in a sense to be outlined. Section 5.3 contains a discussion of an interesting nonuniqueness property of the method, while, in section 5.4, the possibility of an 'optimal' LS method of model reduction is explored.

## 5.2 Least-squares Moment Matching

In section 4.3 an account was given of the model reduction technique of LS moment matching and it can be seen from the description given in section 4.4 that the method of Aguirre (1992) is easily adapted to this special case. The former methods (Shoji *et al* 1985, Lucas and Beat 1990) are what are referred to as the *partial* LS method (section 4.4), in that only the denominator coefficients are approximated in the LS sense before the numerator coefficients are derived by exact moment matching. On the other hand, the method of Aguirre (1992) is what is termed the *full* LS method (section 4.4), in that both the numerator and denominator are derived at

the same time in a single generalised inverse operation. Here (Smith and Lucas 1995) it is shown by considering the matrix-vector form of these methods, that, for LS Padé methods involving the use of time moments only or Markov parameters only, the full and partial methods are equivalent.

**Equivalence of Full and Partial Least-squares Methods**

For a reduced $k$th order model given by

$$R(s) = \frac{d_{k-1}s^{k-1} + \cdots + d_1 s + d_0}{s^k + e_{k-1}s^{k-1} + \cdots + e_1 s + e_0} \tag{5.1}$$

which is derived from the first $2k + t$ time moments of $G(s)$, where $t$ is the number of extra time moments used to extend the exact Padé method, the following set of equations are formed (Aguirre 1992) by equating (3.7) and (5.1)

$$\begin{aligned}
d_0 &= e_0 c_0 \\
d_1 &= e_0 c_1 + e_1 c_0 \\
&\vdots \\
d_{k-1} &= e_0 c_{k-1} + e_1 c_{k-2} + \cdots + e_{k-1} c_0 \\
0 &= e_0 c_k + e_1 c_{k-1} + \cdots + e_{k-1} c_1 + c_0 \\
&\vdots \\
0 &= e_0 c_{2k+t-1} + e_1 c_{2k+t-2} + \cdots + e_{k+t-1} c_k + c_{k+t-1}
\end{aligned} \tag{5.2}$$

Clearly, these may be expressed in the matrix-vector from of

$$A\mathbf{x} = \mathbf{b}$$

or in partitioned form as

$$\left[\begin{array}{c|c} I_k & C_0 \\ \hline \Phi & C_1 \end{array}\right]\left[\begin{array}{c} \mathbf{d} \\ \hline \mathbf{e} \end{array}\right]=\left[\begin{array}{c} \mathbf{0} \\ \hline \mathbf{q} \end{array}\right] \tag{5.3}$$

where $I_k$ is the $k \times k$ identity matrix, $\Phi$ is the $(k+t)\times k$ null matrix and

$$\mathbf{d}=[d_0 \quad d_1 \quad \ldots \quad d_{k-1}]^T$$

$$\mathbf{e}=[e_0 \quad e_1 \quad \ldots \quad e_{k-1}]^T$$

$$\mathbf{q}=[c_0 \quad c_1 \quad \ldots \quad c_{k+t-1}]^T$$

$$C_0=\begin{bmatrix} -c_0 & 0 & \cdots & 0 \\ -c_1 & -c_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -c_{k-1} & -c_{k-2} & \cdots & -c_0 \end{bmatrix}$$

$$C_1=\begin{bmatrix} -c_k & -c_{k-1} & \cdots & -c_1 \\ -c_{k+1} & -c_k & \cdots & -c_2 \\ \vdots & \vdots & \vdots & \vdots \\ -c_{2k+t-1} & -c_{2k+t-2} & \cdots & -c_{k+t} \end{bmatrix}$$

Here it is also noticed that the corresponding matrix-vector equation for the partial method (Shoji *et al* 1985, Lucas and Beat 1990), to determine the denominator, can be written simply as

$$C_1\mathbf{e}=\mathbf{q} \tag{5.4}$$

Now the LS solution of (5.3) is obtained from

$$A^T A\mathbf{x}=A^T\mathbf{b}$$

and notice that

$$A^T A=\left[\begin{array}{c|c} I_k & \Phi^T \\ \hline C_0^T & C_1^T \end{array}\right]\left[\begin{array}{c|c} I_k & C_0 \\ \hline \Phi & C_1 \end{array}\right]$$

$$=\left[\begin{array}{c|c} I_k & C_0 \\ \hline C_0^T & C_0^T C_0+C_1^T C_1 \end{array}\right]$$

and

$$A^T\mathbf{b} = \left[\begin{array}{c|c} I_k & \Phi^T \\ \hline C_0^T & C_1^T \end{array}\right]\left[\begin{array}{c} \mathbf{0} \\ \hline \mathbf{q} \end{array}\right] = \left[\begin{array}{c} \mathbf{0} \\ \hline C_1^T\mathbf{q} \end{array}\right]$$

which then yields the LS solution from

$$\left[\begin{array}{c|c} I_k & C_0 \\ \hline C_0^T & C_0^TC_0 + C_1^TC_1 \end{array}\right]\left[\begin{array}{c} \mathbf{d} \\ \hline \mathbf{e} \end{array}\right] = \left[\begin{array}{c} \mathbf{0} \\ \hline C_1^T\mathbf{q} \end{array}\right]$$

This is equivalent to solving the two blocks of equations

$$\mathbf{d} + C_0\mathbf{e} = \mathbf{0} \tag{5.5}$$

and

$$C_0^T\mathbf{d} + \left(C_0^TC_0 + C_1^TC_1\right)\mathbf{e} = C_1^T\mathbf{q} \tag{5.6}$$

Substituting for $\mathbf{d}$ in (5.6) using (5.5) clearly gives

$$C_1^TC_1\mathbf{e} = C_1^T\mathbf{q}$$

which also gives the LS solution of (5.4) for the partial method. It is further noticed that (5.5) gives the numerator coefficient vector $\mathbf{d}$, once $\mathbf{e}$ has been found, which is equivalent to using the first $k$ of (5.2) to match the first $k$ time moments, as in the partial method. Hence, models obtained by the full and partial LS methods are identical.

This important result allows insight into how the LS moment matching method actually approximates the full system. Notice that the technique may now be thought of as the two-stage process of denominator calculation, by solving (5.4) in an LS sense, and then numerator calculation by simple substitution into (5.5). In solving (5.4) it is seen that the Euclidean norm of the vector $\left(C_1\mathbf{e} - \mathbf{q}\right)$ is minimised, and $\mathbf{q}$ contains the first $k + t$ time moments of the system, whereas $C_1\mathbf{e}$ is an *estimate* of the time moments of the corresponding reduced model because $C_1$ contains the time

moment parameters of the *full* system and not those of the reduced model. Hence, it is clear that the method minimises the index $J$, where

$$J^2 = \sum_{i=0}^{k+t-1} \left(c_i - c_i^*\right)^2$$

for the denominator, where $c_i^*$ $(i = 0,\ 1,\ \ldots\ ,\ k+t-1)$ are only *estimates* of the time moments of the reduced model (as given by $C_1\mathbf{e}$), and then matches the first $k$ time moments *exactly* for the numerator coefficients. This differs fundamentally from the implicit assumption in earlier work that the method minimises the sum of the squared differences between the first $2k + t$ time moments of the full and reduced models.

Finally, it is noticed that the equivalence of the full and partial approaches means that, from the point of view of computational efficiency, the partial approach is to be preferred because of smaller dimensional matrix calculation and consequently less propagation of numerical errors when performing the operations in the LS solutions.

## 5.3 A Nonuniqueness Property

In the light of the two-stage analysis of the LS moment matching carried out in the previous section another property of the method will now be proved (Lucas and Smith, 1995) which has relevance to choosing a subset of system time moments to be used in the LS reduced denominator calculation. It is shown that setting a different denominator or numerator coefficient equal to unity (for the free parameter choice) produces a different reduced order model, unlike the exact Padé method. It is also shown that for the denominator case the moment matching property still holds, whereas in the numerator case this is not so.

### Unity coefficient in denominator

Suppose a reduced $k$th order model given by the transfer function

$$R(s) = \frac{d_{k-1}s^{k-1} + \cdots + d_1 s + d_0}{e_k s^k + e_{k-1}s^{k-1} + \cdots + e_1 s + e_0}$$

is to be found by the LS Padé method using the first $2k + t$ time moments from the full system transfer function $G(s)$, where

$$G(s) = c_0 + c_1 s + c_2 s^2 + \ldots$$

and the $c_i$ are the time moments. Most authors naturally choose $e_k$ equal to unity, leaving $2k$ parameters to be determined in the reduced model $R(s)$. However, if instead another denominator coefficient, say $e_j$ $(0 \le j \le k-1)$, is chosen to be unity, then the Padé equations to be solved in an LS sense become, by matching the expansions of $R(s)$ and $G(s)$ about $s = 0$,

$$\begin{aligned}
d_0 &= e_0 c_0 \\
d_1 &= e_0 c_1 + e_1 c_0 \\
&\vdots \\
d_{j-1} &= e_0 c_{j-1} + e_1 c_{j-2} + \ldots + e_{j-1} c_0 \\
d_j &= e_0 c_j + e_1 c_{j-1} + \ldots + e_{j-1} c_1 + c_0 \\
d_{j+1} &= e_0 c_{j+1} + e_1 c_j + \ldots + e_{j-1} c_2 + c_1 + e_{j+1} c_0 \\
&\vdots \\
d_{k-1} &= e_0 c_{k-1} + e_1 c_{k-2} + \ldots + c_{k-j-1} + \ldots + e_{k-1} c_0 \\
0 &= e_0 c_k + e_1 c_{k-1} + \ldots + c_{k-j} + \ldots + e_k c_0 \\
&\vdots \\
0 &= e_0 c_{2k+t-1} + e_1 c_{2k+t-2} + \ldots + c_{2k+t-j-1} + \ldots + e_k c_{k+t-1}
\end{aligned} \tag{5.7}$$

These equations may be expressed in the partitioned matrix-vector form of

$$\left[\begin{array}{c|c} I_k & C_0 \\ \hline \Phi & C_1 \end{array}\right] \left[\begin{array}{c} \mathbf{d} \\ \hline \mathbf{e} \end{array}\right] = \left[\begin{array}{c} \mathbf{p} \\ \hline \mathbf{q} \end{array}\right] \tag{5.8}$$

very similar to (5.3), where $I_k$ , $\Phi$ and $\mathbf{d}$ are defined as before, but

$$\mathbf{e} = \begin{bmatrix} e_0 & e_1 & \cdots & e_{j-1} & e_{j+1} & \cdots & e_k \end{bmatrix}^T$$

$$\mathbf{p} = \begin{bmatrix} 0 & \cdots & 0 & c_0 & c_1 & \cdots & c_{k-j-1} \end{bmatrix}^T$$

$$\mathbf{q} = \begin{bmatrix} c_{k-j} & c_{k-j+1} & \cdots & c_{2k+t-j-1} \end{bmatrix}^T$$

$$C_0 = \begin{bmatrix} -c_0 & \cdots & \cdots & \cdots & 0 & 0 & 0 & 0 \\ -c_1 & -c_0 & \cdots & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & & \vdots & \vdots & \vdots & \vdots \\ -c_{j-1} & -c_{j-2} & \cdots & -c_0 & \cdots & \cdots & \cdots & 0 \\ -c_j & -c_{j-1} & \cdots & -c_1 & 0 & \cdots & \cdots & 0 \\ -c_{j+1} & -c_j & \cdots & -c_2 & -c_0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \ddots & \ddots & \vdots \\ -c_{k-1} & -c_{k-2} & \cdots & -c_{k-j} & -c_{k-j-2} & \cdots & -c_0 & 0 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} -c_k & -c_{k-1} & \cdots & -c_{k-j+1} & -c_{k-j-1} & \cdots & -c_0 \\ -c_{k+1} & -c_k & \cdots & -c_{k-j+2} & -c_{k-j} & \cdots & -c_1 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ -c_{2k+t-1} & -c_{2k+t-2} & \cdots & -c_{2k+t-j} & -c_{2k+t-j-2} & \cdots & -c_{k+t-1} \end{bmatrix}$$

Performing the LS solution (Lucas and Smith 1995) on (5.8) gives

$$\left[\begin{array}{c|c} I_k & C_0 \\ \hline C_0^T & C_0^T C_0 + C_1^T C_1 \end{array}\right] \left[\begin{array}{c} \mathbf{d} \\ \hline \mathbf{e} \end{array}\right] = \left[\begin{array}{c} \mathbf{p} \\ \hline C_0^T \mathbf{p} + C_1^T \mathbf{q} \end{array}\right]$$

which is equivalent to solving the two matrix equations

$$\mathbf{d} + C_0 \mathbf{e} = \mathbf{p} \tag{5.9}$$

and

$$C_0^T \mathbf{d} + \left(C_0^T C_0 + C_1^T C_1\right)\mathbf{e} = C_0^T \mathbf{p} + C_1^T \mathbf{q} \tag{5.10}$$

Substituting for $\mathbf{d}$ from (5.9) into (5.10) gives

$$C_1^T C_1 \mathbf{e} = C_1^T \mathbf{q} \tag{5.11}$$

which is also seen to be the LS equation that yields the denominator coefficient vector $\mathbf{e}$ directly. Substituting this into (5.9) gives the numerator vector $\mathbf{d}$. Equation (5.9) is

simply another way of writing the first $k$ of (5.7), thus ensuring that the first $k$ time moments of $G(s)$ are retained in $R(s)$. This demonstrates the expected result that when $e_j = 1$, for arbitrary $j$, the full LS Padé method can be decomposed into a two-staged partial method.

This allows further insight into the error criterion used in the approximation. Equation (5.9) implies that the Euclidean norm of the vector $C_1\mathbf{e} - \mathbf{q}$ is minimised. However, $\mathbf{q}$ contains the system time moments $c_i$, $i = k-j, \ldots, 2k+t-j-1$, as its components, whereas $C_1\mathbf{e}$ represents the corresponding *estimates* $c_i^*$ for the reduced model as observed in the previous section. Hence the error index $J$ minimised for the denominator calculation is given by

$$J^2 = \sum_{i=k-j}^{2k+t-j-1} \left(c_i - c_i^*\right)^2$$

It is observed that the summation in the index always involves $k+t$ successive time moments, but these vary as $j$ varies. At the extreme values, $j = 0$ is associated with the last $k+t$ time moments used in the Padé equations, and $j = k$ is associated with the first $k+t$. Although the first $k$ system time moments are always preserved in the reduced model, $k+1$ different reduced models will result by setting a different denominator coefficient equal to unity.

**Example 5.1**

To illustrate this interesting result, consider the fourth-order transfer function (Aguirre 1992)

$$G(s) = \frac{267s^3 + 527s^2 + 385s + 100}{s^4 + 4s^3 + 6s^2 + 4s + 1}$$

which has an expansion about $s = 0$ given by

$$G(s) = 100 - 15s - 13s^2 + 9s^3 + 2s^4 + 5s^5 - 55s^6 + 173s^7 + \ldots$$

Reducing $G(s)$ to second-order by the LS Padé method with $t = 4$ and $e_j = 1$ ($j = 0, 1, 2$) gives

$$j = 0, \qquad R_0(s) = \frac{284.926s + 100}{0.539487s^2 + 2.99926s + 1}$$

with

$$I_{rel} = 28.25\% \qquad \text{and} \qquad J_{rel} = 14.4\%$$

$$j = 1, \qquad R_1(s) = \frac{95.576s + 29.4926}{0.177451s^2 + s + 0.294926}$$

with

$$I_{rel} = 30.1\% \qquad \text{and} \qquad J_{rel} = 15.1\%$$

and

$$j = 2, \qquad R_2(s) = \frac{251.464s + 78.7052}{s^2 + 2.6327s + 0.787052}$$

$$I_{rel} = 1.37\% \qquad \text{and} \qquad J_{rel} = 7.53\%$$

It is easily verified that the first two time moments of the system are retained in all three reduced models, but that these models vary according to which denominator coefficient is set equal to unity. It is interesting that $R_2(s)$ is the best model in terms of integral square error values and that this uses the first six system time moments of $G(s)$ in the LS approximation of the denominator. Observation of these values for the other two models shows that prediction of the value of $j$ that gives the 'best' model is not easy and needs to be treated with caution. A comparison of the impulse and step responses of these reduced models with those of the full system is given in figures 5.1 and 5.2 respectively at the end of this section.

**Unity coefficient in numerator**

It is interesting to see what happens when these ideas are now extended to the numerator coefficients in the reduced LS model. If one of the numerator coefficients of $R(s)$, say $d_j$ $(0 \le j \le k-1)$, is chosen to be unity, then again using $t$ extra time moments the Padé equations to be solved in an LS sense become

$$\begin{aligned}
&d_0 = e_0 c_0 \\
&\vdots \\
&d_{j-1} = e_{j-1} c_0 + \cdots + e_0 c_{j-1} \\
&1 = e_j c_0 + e_{j-1} c_1 + \cdots + e_0 c_j \\
&d_{j+1} = e_{j+1} c_0 + \cdots + e_0 c_{j+1} \\
&\vdots \\
&d_{k-1} = e_{k-1} c_0 + e_{k-2} c_1 + \cdots + e_0 c_{k-1} \\
&0 = e_k c_0 + e_{k-1} c_1 + \cdots + e_0 c_k \\
&\vdots \\
&0 = e_k c_{k+t-1} + e_{k-1} c_{k+t} + \cdots + e_0 c_{2k+t-1}
\end{aligned} \tag{5.12}$$

where the equation for $d_j$, set to unity, may be placed last in the set of numerator coefficient equations. This is so that (5.12) may be expressed in the partitioned matrix form of

$$\left[\begin{array}{c|c} I_{k-1} & C_0 \\ \hline \Phi & C_1 \end{array}\right]\left[\begin{array}{c} \mathbf{d} \\ \hline \mathbf{e} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \mathbf{q} \end{array}\right] \tag{5.13}$$

where the various partitions are of different orders to those in (5.8). $I_{k-1}$ is the $(k-1)\times(k-1)$ identity matrix, $\Phi$ is the $(k+t+1)\times(k-1)$ null matrix,

$$\mathbf{d} = \begin{bmatrix} d_0 & \cdots & d_{j-1} & d_{j+1} & \cdots & d_{k-1} \end{bmatrix}^T \qquad \mathbf{e} = \begin{bmatrix} e_0 & e_1 & \cdots & e_k \end{bmatrix}^T$$

$$\mathbf{q} = \begin{bmatrix} -1 & 0 & \cdots & 0 \end{bmatrix}^T$$

$$C_0 = \begin{bmatrix} -c_0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ -c_1 & -c_0 & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ -c_{j-1} & -c_{j-2} & \cdots & -c_0 & \cdots & 0 & 0 \\ -c_{j+1} & -c_j & \cdots & -c_0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ -c_{k-1} & -c_{k-2} & \cdots & 0 & \cdots & -c_0 & 0 \end{bmatrix}$$

and

$$C_1 = \begin{bmatrix} -c_j & -c_{j-1} & \cdots & 0 & 0 \\ -c_k & -c_{k-1} & \cdots & -c_1 & -c_0 \\ -c_{k+1} & -c_k & \cdots & -c_2 & -c_1 \\ \vdots & \vdots & & \vdots & \vdots \\ -c_{2k+t-1} & -c_{2k+t-2} & \cdots & -c_{k+t} & -c_{k+t-1} \end{bmatrix}$$

Proceeding in the usual way, the LS solution of (5.13) gives

$$\left[\begin{array}{c|c} I_{k-1} & C_0 \\ \hline C_0^T & C_0^T C_0 + C_1^T C_1 \end{array}\right]\left[\begin{array}{c} \mathbf{d} \\ \hline \mathbf{e} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline C_1^T \mathbf{q} \end{array}\right]$$

which is equivalent to solving the two matrix-vector equations

$$\mathbf{d} + C_0 \mathbf{e} = 0 \tag{5.14}$$

$$C_0^T \mathbf{d} + \left(C_0^T C_0 + C_1^T C_1\right)\mathbf{e} = C_1^T \mathbf{q} \tag{5.15}$$

As expected, substituting for **d** from (5.14) into (5.15) gives a result similar to (5.11), i.e.

$$C_1^T C_1 \mathbf{e} = C_1^T \mathbf{q}$$

from which it follows that when $d_j = 1$, for arbitrary $j$, the full LS Padé method can still be decomposed into a two-stage partial LS method.

In this case, however, it is noticed that the Euclidean norm of the vector $C_1\mathbf{e} - \mathbf{q}$ which is minimised does not have the same meaning as that of (5.11). It is clear that the vector **q** does not contain the full system time moments and, therefore, the error index minimised *cannot* be characterised as the sum of the squares of the differences between the full system time moments and estimates of the reduced

system time moments. It is further interesting to notice that the numerator coefficients, given by (5.14), will ensure matching of only the first $j$ time moments between the full and reduced models. This is a direct consequence of using the constraint equation, with $d_j = 1$, in the LS calculation of the denominator.

**Example 5.2**

Returning to the same fourth-order transfer function as for example 5.1 and reducing $G(s)$ to second-order by the LS Padé method with $t = 4$ and $d_j = 1$ $(j = 0,1)$ gives

$$j = 0, \qquad R_0(s) = \frac{2.05997s + 1}{0.0039004s^2 + 0.0216842s + 0.00722986}$$

with

$$I_{rel} = 27.35\%$$

and

$$j = 1, \qquad R_1(s) = \frac{s + 0.294485}{0.00178149s^2 + 0.0100457s + 0.00294485}$$

with

$$I_{rel} = 34.7\% \qquad J_{rel} = 16.4\%$$

Examination of the sequences of time moments for these two reduced models verifies that only the first $j$ time moments are retained in the reduced models $R_0(s)$ retains no time moments and $R_1(s)$ retains only the first. It is not surprising, therefore, that the relative integral square errors show that the reduced models produced in this way are not a significant improvement on any of those produced by setting an arbitrary denominator coefficient equal to unity. Comparisons of the impulse and step responses of these reduced models with those of the full system are given in figures 5.3 and 5.4 respectively.

**Fig 5.1 Unity Coefficient in Denominator**

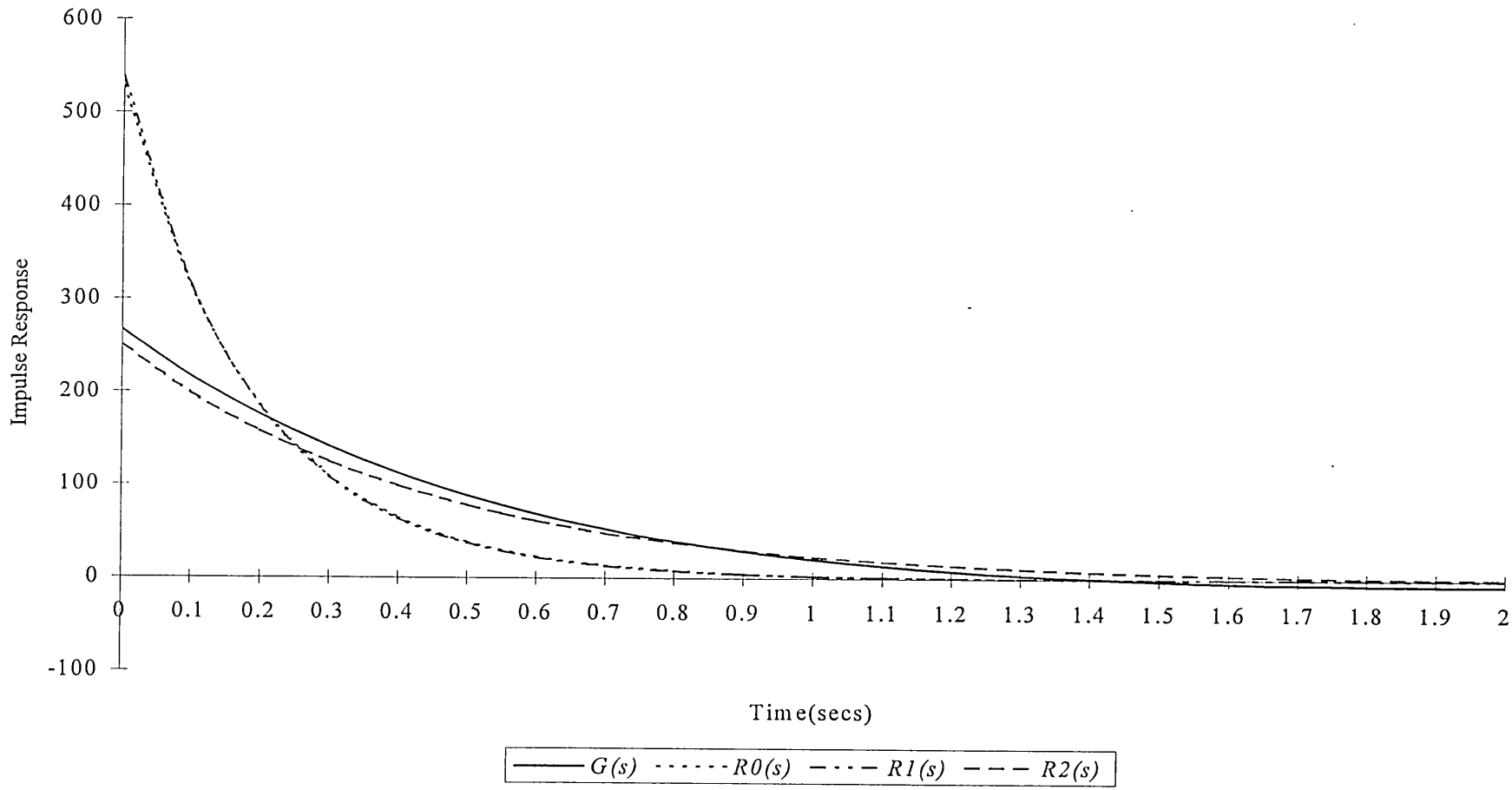


**Fig 5.2 Unity Coefficient in Denominator**

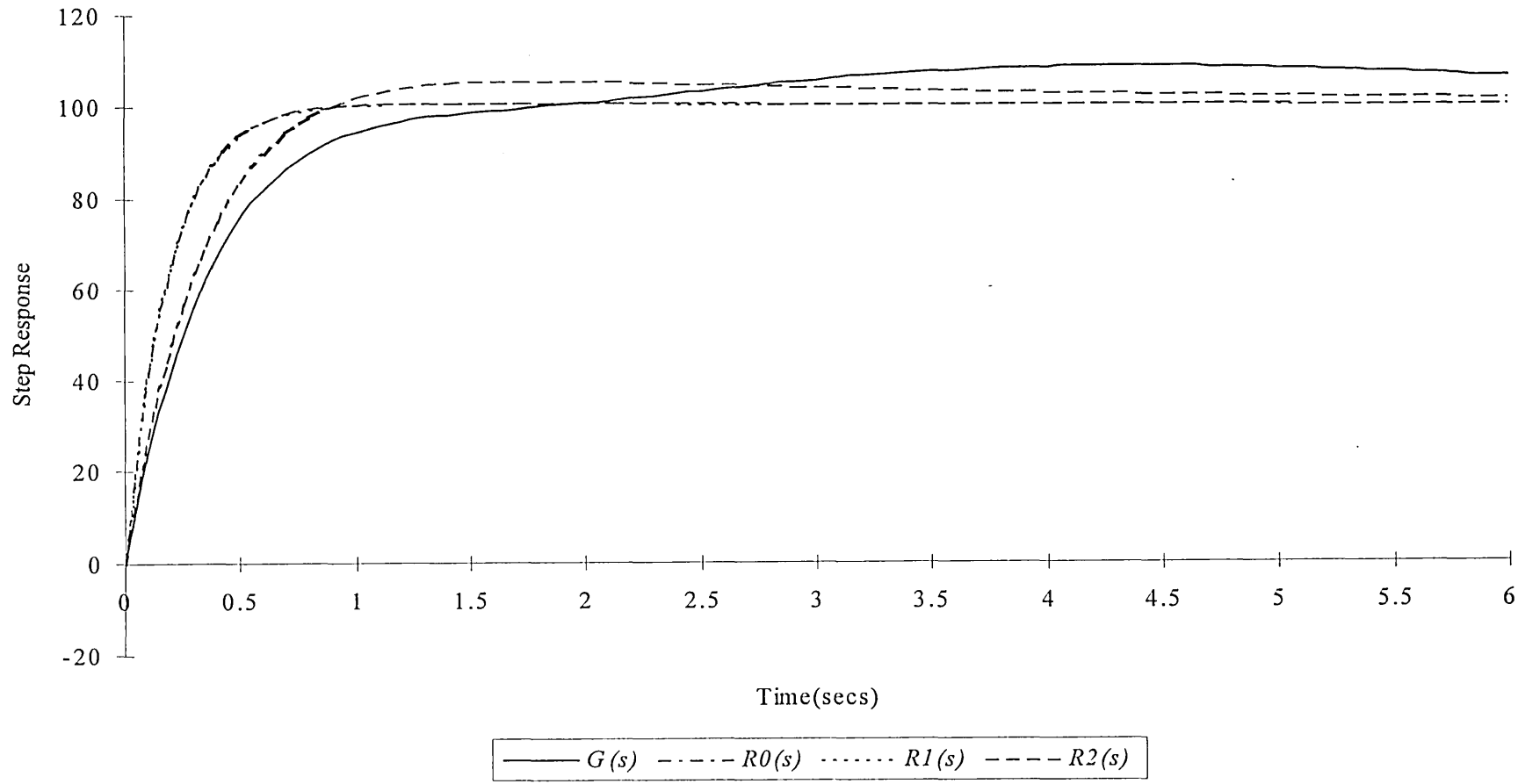


**Fig 5.3 Unity Coefficent in Numerator**

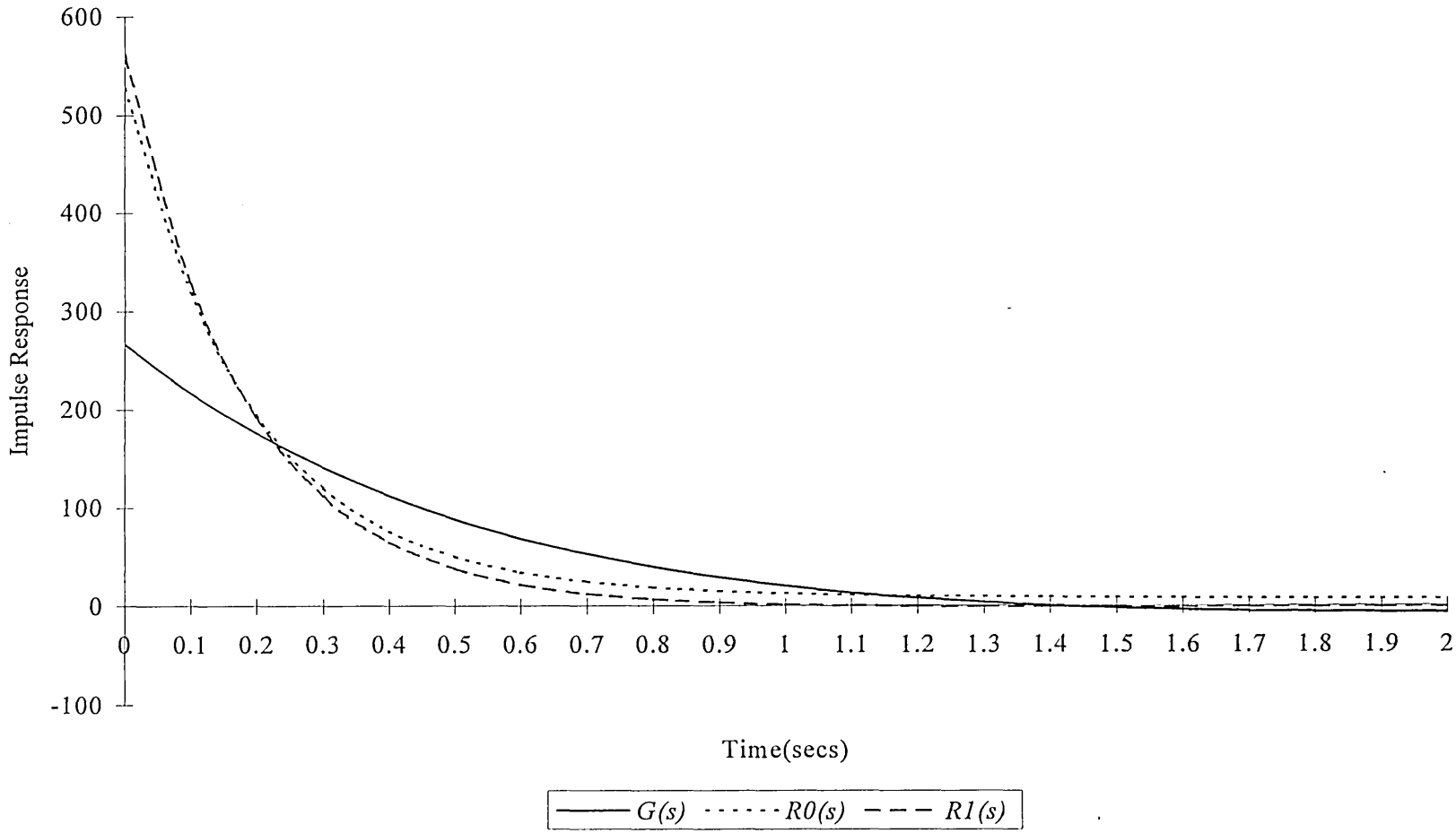
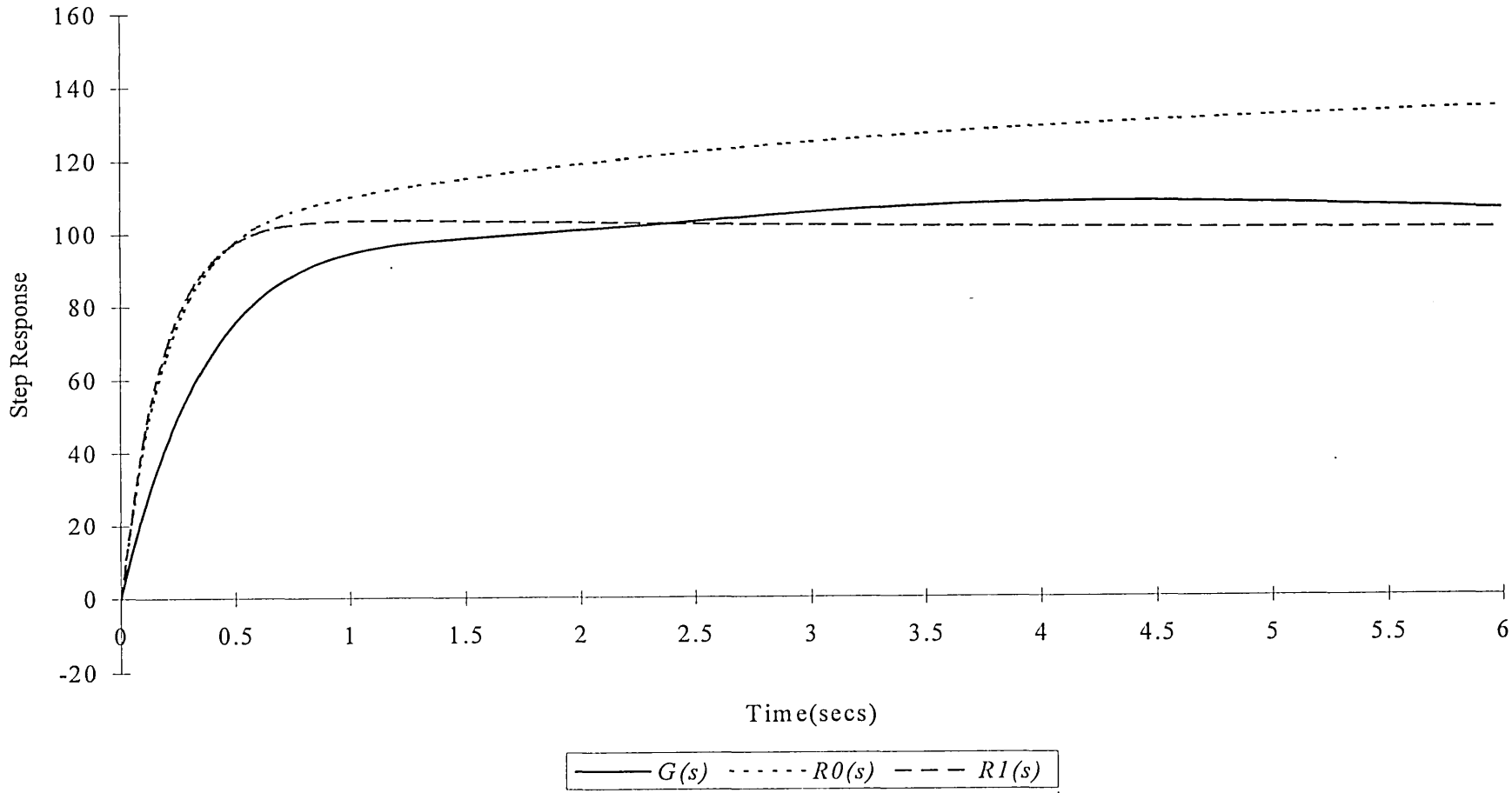


**Fig 5.4 Unity Coefficent in Numerator**



## 5.4 Optimal Least-squares Method

In section 5.2 it is seen that the LS method described so far is not truly optimal in the sense that it minimizes the Euclidean norm of the vector $(C_1\mathbf{e}-\mathbf{q})$ which contains only *estimates* of the time moments of the reduced model. This point leads naturally to a consideration of the possibility of adapting the LS method so as to achieve 'true' optimization. This would involve minimizing an error index based on the sum of the squares of the differences of the actual time moments of the full and reduced systems respectively.

Although such an LS calculation cannot be performed with ease directly, an iterative procedure can be performed which, in the event of convergence, may be regarded as 'optimizing' the LS reduced order model produced. This is achieved by carrying out repeated LS Padé approximations using the reduced model's actual time moment information to derive the next reduced model iteration in the sequence. This should lead, in the event of convergence, to an 'optimal' model.

In the first stage of the optimal method proposed, LS moment matching is performed on $G(s)$ by solving (5.9) where

$$\mathbf{e}=\begin{bmatrix} e_1 & e_2 & \cdots & e_k \end{bmatrix}^T \qquad \mathbf{p}=\begin{bmatrix} c_0 & c_1 & \cdots & c_{k-1} \end{bmatrix}^T$$

$$\mathbf{q}=\begin{bmatrix} c_k & c_{k+1} & \cdots & c_{2k+l-1} \end{bmatrix}^T$$

$$C_0=\begin{bmatrix} 0 & 0 & \cdots & 0 \\ -c_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -c_{k-2} & -c_{k-3} & \cdots & 0 \end{bmatrix}$$

$$C_1=\begin{bmatrix} -c_{k-1} & -c_{k-2} & \cdots & -c_0 \\ -c_k & -c_{k-1} & \cdots & -c_1 \\ \vdots & \vdots & & \vdots \\ -c_{2k+l-2} & -c_{2k+l-3} & \cdots & -c_{k+l-1} \end{bmatrix}$$

resulting in a $k$th order reduced model

$$R_1(s) = \frac{d_{k-1}s^{k-1} + \cdots + d_1 s + d_0}{e_k s^k + e_{k-1}s^{k-1} + \cdots + e_1 s + 1}.$$

It is noted that the option of setting an arbitrary denominator coefficient to unity, as in section 5.3, has been applied. In this case the coefficient $e_0$ has been chosen so that for consistency at each iteration it is the first $k$ time moments which are matched exactly to calculate the numerator and it is the last $k + t$ time moments which are approximated in the LS sense. At each iteration the error index minimized is $J$ given by

$$J^2 = \sum_{i=k}^{2k+t-1} \left(c_i - c_i^*\right)^2$$

The $c_i$ are the time moments of the full system and the $c_i^*$ are now *estimates* of the new reduced model's time moments using the time moment information from the previous iterate.

The process is then continued by taking $R_1(s)$ as a first iterate, the LS Padé approximation being repeated with the solution of

$$C_1^T C_1 \mathbf{e} = C_1^T \mathbf{q}$$

to give a second iterate $R_2(s)$ where $\mathbf{q}$ still contains the full system's time moments, but $C_1$ now contains the *actual* time moments of $R_1(s)$. In this step, therefore, the vector $C_1\mathbf{e}$ contains *estimates* of the time moments of the second iterate $R_2(s)$ derived from the time moment information of the first iterate $R_1(s)$. This process may be repeated for as many iterations as desired in an attempt to converge to an improved LS approximation to the full system.

The results produced by this iterative procedure have proved disappointing with no convergence taking place in many examples and little or no improvement over the accuracy of the first iterate being achieved even in those examples where convergence did result. These points are demonstrated in the following example which has been considered already in Chapters 3 and 4.

Consider

$$G(s)=\frac{s^5+17.5s^4+111s^3+314.5s^2+388s+168}{s^6+15s^5+93s^4+307s^3+562s^2+562s+260}$$

to which the proposed optimal LS method was applied. The results were as follows with the first iterate being the third-order model produced by LS moment matching with the constant denominator term set to unity

$$R_1(s)=\frac{0.994716s^2+1.44666s+0.646152}{1.07454s^3+1.8394s^2+2.0909s+1}$$

with

$$I_{rel}=31.6\% \qquad \text{and} \qquad J_{rel}=15.9\%$$

This was followed by a second iterate

$$R_2(s)=\frac{1.09772s^2+1.54053s+0.646152}{1.19582s^3+1.97732s^2+2.23617s+1}$$

The relative ISE values for this being

$$I_{rel}=31.8\% \qquad \text{and} \qquad J_{rel}=16.3\%$$

A third iterate

$$R_3(s)=\frac{0.922808s^2+1.26972s+0.646152}{0.950057s^3+1.76864s^2+1.81707s+1}$$

gives

$$I_{rel} = 34.3\% \qquad \text{and} \qquad J_{rel} = 19\%$$

This is typical of the sort of disappointing results when applying the method. Where there is convergence to a value there is often a deterioration in the accuracy rather than an improvement and any improvement that does occur is usually minimal.

## 5.5 Generalised LS Padé Approximation

In this section consideration is given to the generalised LS method for the reduction of continuous-time systems and it will be established that the two-stage analysis given in section 5.2 may be used to prove interesting properties in this more general case. It further serves to put various LS methods into a unified theoretical framework (Smith and Lucas 1996).

Suppose that a reduced $k$th order model is to be found by the LS Padé method using the system information from the first $2k + t$ time moments and the first $r$ Markov parameters, where, for initial development of the theory, it is assumed that $t > 0$ and $r < k$ ($r \geq k$ is seen to be a special case and is considered later). Lucas and co-workers (1990, 1991) discuss how typical values of $t$ and $r$ might be chosen and an illustration is given later in this section. To incorporate the time moment information, the following set of equations is formed (Lucas and Munro 1991, Aguirre 1992) by directly equating the coefficients of powers of $s$ in the expansions of $R(s)$ and $G(s)$ respectively about $s = 0$ as far as $s^{2k+t-1}$; that is

$$
\begin{aligned}
d_0 &= e_0 c_0 \\
d_1 &= e_1 c_0 + e_0 c_1 \\
&\vdots \\
d_{k-1} &= e_{k-1} c_0 + \cdots + e_0 c_{k-1} \\
0 &= c_0 + e_{k-1} c_1 + \cdots + e_0 c_k \\
0 &= c_1 + e_{k-1} c_2 + \cdots + e_0 c_{k+1} \\
&\vdots \\
0 &= c_{k+t-1} + e_{k-1} c_{k+t} + \cdots + e_0 c_{2k+t-2}
\end{aligned}
\tag{5.16}
$$

Also, to use the Markov parameter information, the following set of equations is formed by matching coefficients of powers of $s^{-1}$ in the expansions of $R(s)$ and $G(s)$ about $s = \infty$ as far as $s^{-r}$; that is

$$
\begin{aligned}
d_{k-1} &= m_1 \\
d_{k-2} &= m_1 e_{k-1} + m_2 \\
&\vdots \\
d_{k-r} &= m_1 e_{k-r+1} + m_2 e_{k-r+2} + \cdots + m_r
\end{aligned}
\tag{5.17}
$$

**Full LS Method**

Equations (5.16) and (5.17) may now be expressed together in the convenient matrix-vector form

$$A\mathbf{x} = \mathbf{b}$$

which in partitioned form is

$$
\left[\begin{array}{c|c} \Phi & C_1 \\ \hline I_k & C_0 \\ \hline \lambda & M_0 \end{array}\right]
\left[\begin{array}{c} \mathbf{d} \\ \hline \mathbf{e} \end{array}\right]
=
\left[\begin{array}{c} \mathbf{p} \\ \hline \mathbf{0} \\ \hline \mathbf{q} \end{array}\right]
\tag{5.18}
$$

where $I_k$ is the $k \times k$ identity matrix, $\Phi$ is the $(k+t) \times k$ null matrix,

$$\mathbf{d} = [d_{k-1} \quad d_{k-2} \quad \ldots \quad d_0]^T \qquad \mathbf{e} = [e_{k-1} \quad e_{k-2} \quad \ldots \quad e_0]^T$$

$$\mathbf{p} = \begin{bmatrix} c_{k+t-1} & c_{k+t-2} & \cdots & c_0 \end{bmatrix}^T \qquad \mathbf{q} = \begin{bmatrix} m_1 & m_2 & \cdots & m_r \end{bmatrix}^T$$

$$C_0 = \begin{bmatrix} -c_0 & -c_1 & \cdots & -c_{k-1} \\ & -c_0 & \cdots & -c_{k-2} \\ & 0 & \ddots & \vdots \\ & & & -c_0 \end{bmatrix} \qquad C_1 = \begin{bmatrix} -c_{k+t} & -c_{k+t+1} & \cdots & -c_{2k+t-1} \\ \vdots & \vdots & & \vdots \\ -c_2 & -c_3 & \cdots & -c_{k+1} \\ -c_1 & -c_2 & \cdots & -c_k \end{bmatrix}$$

$$\leftarrow k-r \rightarrow$$

$$M_0 = \left[\begin{array}{cccc|c} 0 & \cdots & \cdots & 0 & \\ -m_1 & 0 & \cdots & 0 & \\ -m_2 & -m_1 & \ddots & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ -m_{r-1} & -m_{r-2} & \cdots & -m_1 & 0 \end{array}\right]$$

and

$$\lambda = \left[\begin{array}{cccc|c} 1 & 0 & \cdots & 0 & \\ 0 & 1 & \cdots & 0 & \\ \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & \cdots & 1 & \end{array}\right]$$

$$\leftarrow \quad r \quad \rightarrow \quad \leftarrow k-r \rightarrow$$

The LS solution (Aguirre 1992) of (5.18) is obtained from

$$A^T A \mathbf{x} = A^T \mathbf{b}$$

which in partitioned form is

$$\left[\begin{array}{c|c} I_k + \lambda^T \lambda & C_0 + \lambda^T M_0 \\ \hline C_0^T + M_0^T \lambda & C_1^T C_1 + C_0^T C_0 + M_0^T M_0 \end{array}\right] \left[\begin{array}{c} \mathbf{d} \\ \hline \mathbf{e} \end{array}\right] = \left[\begin{array}{c} \lambda^T \mathbf{q} \\ \hline C_1^T \mathbf{p} + M_0^T \mathbf{q} \end{array}\right] \tag{5.19}$$

This in turn is seen to be equivalent to solving the two matrix-vector equations

$$\left(I_k + \lambda^T \lambda\right)\mathbf{d} + \left(C_0 + \lambda^T M_0\right)\mathbf{e} = \lambda^T \mathbf{q} \tag{5.20}$$

and

$$\left(C_0^T + M_0^T \lambda\right)\mathbf{d} + \left(C_1^T C_1 + C_0^T C_0 + M_0^T M_0\right)\mathbf{e} = C_1^T \mathbf{p} + M_0^T \mathbf{q} \tag{5.21}$$

Solving (5.20) for **d** gives

$$\mathbf{d} = \left(I_k + \lambda^T \lambda\right)^{-1} \left\{\lambda^T \mathbf{q} - \left(C_0 + \lambda^T M_0\right)\mathbf{e}\right\} \tag{5.22}$$

and substituting this into (5.21), using the fact that $\left(C_0^T + M_0^T \lambda\right)$ is identically equal to $\left(C_0 + \lambda^T M_0\right)^T$, gives

$$\left\{-\left(C_0 + \lambda^T M_0\right)^T \left(I_k + \lambda^T \lambda\right)^{-1} \left(C_0 + \lambda^T M_0\right) + C_1^T C_1 + C_0^T C_0 + M_0^T M_0\right\}\mathbf{e}$$

$$= C_1^T \mathbf{p} + M_0^T \mathbf{q} - \left(C_0 + \lambda^T M_0\right)^T \left(I_k + \lambda^T \lambda\right)^{-1} \lambda^T \mathbf{q} \tag{5.23}$$

Hence, it is seen that the full generalised LS method (Aguirre 1992) may be implemented as a *two-stage* process. The first stage is to solve (5.23) directly for the reduced denominator vector **e**, and the second stage is to substitute these values into (5.22) to calculate the reduced numerator coefficient vector **d**. Once again this analysis in terms of a two-stage process is the key to providing a deeper understanding of the full generalised LS method and its relationship with the various other LS methods.

**Partial Least-Squares Method**

In their partial LS method, Lucas and Munro (1991) use Markov parameters as well as time moments in the LS approximation of the reduced denominator. As such, this will be used to illustrate the typical partial LS approach for denominator calculation. Essentially, this is achieved by eliminating $d_{k-i}$, $i = 1, 2, \ldots, r$, from (5.16) by substituting the expressions for $d_{k-i}$ from (5.17) to produce the matrix equation

$$H\mathbf{e} = \mathbf{g}$$

which in partitioned form is

$$\left[\begin{array}{c} C_1 \\ \hline M_0 - C_{0,r} \end{array}\right]\mathbf{e} = \left[\begin{array}{c} \mathbf{p} \\ \hline \mathbf{q} \end{array}\right] \tag{5.24}$$

where $C_{0,r}$ is the $r \times k$ matrix consisting of the *first* $r$ rows of $C_0$. The LS solution of (5.24) is then obtained from

$$H^T H\mathbf{e} = H^T\mathbf{g}$$

that is from the partitioned matrix equation

$$\left\{C_1^T C_1 + \left(M_0 - C_{0,r}\right)^T \left(M_0 - C_{0,r}\right)\right\}\mathbf{e} = C_1^T\mathbf{p} + \left(M_0 - C_{0,r}\right)^T\mathbf{q} \qquad (5.25)$$

Lucas and Munro (1991) then calculate the reduced numerator coefficients by simply matching the first $k$ system parameters to those of the reduced model by the appropriate $k$ equations from (5.16) and (5.17). For example, if $k$ time moments are to be retained then $\mathbf{d} = -C_0\mathbf{e}$ is used.

At this point, it is of interest to compare the denominator obtained by (5.25) to that of the full LS method given by (5.23). From the definitions of $C_{0,r}$ and $\lambda$ it is clear that

$$C_0 = \left[\begin{array}{c} C_{0,r} \\ \hline C_{0,k-r} \end{array}\right] \quad \text{and} \quad \lambda^T = \left[\begin{array}{cccc} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ \hline & & 0 & \end{array}\right] \begin{array}{l} \uparrow \\ \\ \\ \\ k-r \\ \downarrow \end{array}$$

$$\leftarrow \quad r \quad \rightarrow$$

Therefore, the following identities hold

$$C_0^T C_0 = C_{0,r}^T C_{0,r} + C_{0,k-r}^T C_{0,k-r}$$

$$\lambda^T\lambda = \left[\begin{array}{c|c} I_r & \Phi \\ \hline \Phi & \Phi \end{array}\right]$$

$$C_0 + \lambda^T M_0 = \left[\begin{array}{c} C_{0,r} + M_0 \\ \hline C_{0,k-r} \end{array}\right] \qquad (5.26)$$

$$I_k + \lambda^T \lambda = \left[\begin{array}{c|c} 2I_r & \Phi \\ \hline \Phi & I_{k-r} \end{array}\right]$$

where $C_{0,k-r}$ is the $(k-r)\times k$ matrix consisting of the *last* $k$ - $r$ rows of $C_0$ and the $\Phi$'s are null matrices of appropriate dimension. It is further noticed that since $I_k + \lambda^T \lambda$ is a diagonal matrix this gives

$$\left(I_k + \lambda^T \lambda\right)^{-1} = \left[\begin{array}{c|c} \frac{1}{2} I_r & \Phi \\ \hline \Phi & I_{k-r} \end{array}\right] \tag{5.27}$$

Substituting (5.26) and (5.27) into (5.23) and simplifying gives

$$\left(C_1^T C_1 + \tfrac{1}{2} C_{0,r}^T C_{0,r} - \tfrac{1}{2} C_{0,r}^T M_0 - \tfrac{1}{2} M_0^T C_{0,r} + \tfrac{1}{2} M_0^T M_0\right)\mathbf{e}$$

$$= C_1^T \mathbf{p} + \tfrac{1}{2} M_0^T \mathbf{q} - \tfrac{1}{2} C_{0,r}^T \mathbf{q} \tag{5.28}$$

which can be written in factorised form

$$\left\{C_1^T C_1 + \tfrac{1}{2}\left(M_0 - C_{0,r}\right)^T \left(M_0 - C_{0,r}\right)\right\}\mathbf{e} = C_1^T \mathbf{p} + \tfrac{1}{2}\left(M_0 - C_{0,r}\right)^T \mathbf{q} \tag{5.29}$$

Comparison with (5.25) shows a similarity in the two equations for calculating **e**, but they are not identical. Nevertheless, this comparison provides the clue as to how the full LS method is equivalent to a two-stage partial LS method, which is described in the next section.

**Relationship between the full and partial methods**

The similarity of equations (5.25) and (5.29) suggests that there is a relationship between the full and partial generalised LS methods. Notice that the factorised terms in these equations differ only by a factor of $\frac{1}{2}$. Indeed, by adopting a modified partitioned form for $H$ and **g** in (5.24), given by

$$\left[\begin{array}{c} C_1 \\ \hline \frac{1}{\sqrt{2}}\left(M_0 - C_{0,r}\right) \end{array}\right]\mathbf{e} = \left[\begin{array}{c} \mathbf{p} \\ \hline \frac{1}{\sqrt{2}}\mathbf{q} \end{array}\right] \tag{5.30}$$

a useful equivalence between the results of the full and partial LS methods is established. Using this modification for the LS solution of **e** in (5.30) gives the matrix equation

$$\left\{C_1^T C_1 + \tfrac{1}{2}\left(M_0 - C_{0,r}\right)^T \left(M_0 - C_{0,r}\right)\right\}\mathbf{e} = C_1^T \mathbf{p} + \tfrac{1}{2}\left(M_0 - C_{0,r}\right)^T \mathbf{q} \qquad (5.31)$$

This is seen to be identical to (5.29), which gives **e** when calculated by the full LS method. This shows that by introducing the factor $\frac{1}{\sqrt{2}}$ into the lower partitions of $H$ and **g** as given by (5.30), the partial LS method will yield the same reduced denominator as the full LS method. The significance of this factor will be explained later in terms of the errors minimized by the method.

To complete the equivalence of the full and partial methods, the respective numerators must also be matched. The full LS method numerator calculation has been shown to be equivalent to solving (5.22) which, using the identities given in (5.26) and (5.27), can be written in the partitioned form

$$\mathbf{d} = \begin{bmatrix} \frac{1}{2}\left\{\mathbf{q} - \left(C_{0,r} + M_0\right)\mathbf{e}\right\} \\ \hline -C_{0,k-r}\mathbf{e} \end{bmatrix} \qquad (5.32)$$

It is now clear that the full LS method (Aguirre1992) is equivalent to the two-stage partial LS method, where the denominator calculation is the LS solution of (5.30) followed by the numerator calculation via (5.32).

Given this relationship it is generally preferable to use the partial two-stage LS method, rather than the corresponding full LS method, because of the much-reduced computational effort involved - essentially only $k \times k$ generalized inverses are needed instead of $2k \times 2k$, which can be a significant saving of computation even for relatively low values of $k$.

**Properties of the Method**

Some very interesting properties of the LS method are seen to follow by considering (5.32) in more detail. The lower partition of this vector equation is seen to contain the same information as the first $k$ - $r$ equations of (5.16); that is the reduced denominator coefficients $d_i$, for $i = 0,\ 1,\ 2,\ \ldots\ ,\ k-r-1$, are obtained by straightforward moment matching. In the upper partition, the expression $\mathbf{q} - (C_{0,r} + M_0)\mathbf{e}$ needs careful interpretation. Notice that $-C_{0,r}\mathbf{e}$ is the matrix-vector representation of the expressions for the $d_{k-i}$, $i = 1,\ 2,\ \ldots\ ,\ r$, from (5.16) and, similarly, $-M_0\mathbf{e} + \mathbf{q}$ is the matrix-vector representation for the same $d_{k-i}$ from (5.17). Therefore, the expression $-C_{0,r}\mathbf{e} - M_0\mathbf{e} + \mathbf{q}$ corresponds to the vector with entries $2d_{k-i}$, $i = 1,\ 2,\ \ldots\ ,\ r$, obtained by adding the relevant equations of (5.16) to the equations of (5.17). Hence, the upper partition in (5.32) indicates that the $r$ numerator coefficients $d_{k-i}$, $i = 1,\ 2,\ \ldots\ ,\ r$, are calculated by the *average* of their separate values given by each of the relevant equations of (5.16) and (5.17) respectively; that is, the $r$ time moment and Markov parameter matching equation pairs are *averaged* for calculating these coefficients.

It is noticed that the above arguments apply in the case where model reduction is performed using $t\,(< k)$ time moments and $2k + v$ Markov parameters. In this case, $d_i$, $i = 0,\ 1,\ 2,\ \ldots\ ,\ t-1$, will be obtained by the averaging process while $d_j$, $j = t,\ t+1,\ \ldots\ ,\ k-1$, will be obtained by exact Markov parameter matching.

It is further interesting to note that various options are open in respect of the preservation of system parameters in the LS model reduction process. A choice can be made to use averaging, as described above, for the calculation of any number $r$ of

the numerator coefficients, in which case only the first $k$ - $r$ time moments will be preserved ($r = 0, 1, 2, \ldots, k-1$).

**Special case of $r \geq k$**

For this case the full LS method gives the LS solution of the linear set (5.16) consisting of $2k + t$ equations containing the time moment information and $k + v$ equations containing the Markov parameter information; that is

$$\begin{aligned}
d_{k+1} &= m_1 \\
d_{k-2} &= m_1 e_{k-1} + m_2 \\
&\vdots \\
d_0 &= m_1 e_1 + m_2 e_2 + \ldots + m_k \\
&\vdots \\
0 &= m_2 e_1 + m_3 e_2 + \ldots + m_{k+1} \\
&\vdots \\
0 &= m_{v+1} e_1 + m_{v+2} e_2 + \ldots + m_{k+v}
\end{aligned} \tag{5.33}$$

In matrix-vector form (5.33) may be expressed as

$$A\mathbf{x} = \mathbf{b}$$

which, in this case, has partitioned form

$$\left[\begin{array}{c|c} \Phi_1 & C_1 \\ \hline I_k & C_0 \\ \hline I_k & M_0 \\ \hline \Phi_2 & M_1 \end{array}\right] \left[\begin{array}{c} \mathbf{d} \\ \hline \mathbf{e} \end{array}\right] = \left[\begin{array}{c} \mathbf{p} \\ \hline \mathbf{0} \\ \hline \mathbf{q} \\ \hline \mathbf{u} \end{array}\right] \tag{5.34}$$

In (5.34), $I_k$, $C_0$, $C_1$, $\mathbf{p}$ have the same definitions as in (5.18), $\Phi_1$ and $\Phi_2$ are $(k+t) \times k$ and $v \times k$ null matrices respectively and

$$\mathbf{q} = \begin{bmatrix} m_1 & m_2 & \ldots & m_k \end{bmatrix}^T \qquad \mathbf{u} = \begin{bmatrix} m_{k+1} & m_{k+2} & \ldots & m_{k+v} \end{bmatrix}^T$$

$$M_0 = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ -m_1 & 0 & \cdots & \cdots & 0 \\ -m_2 & -m_1 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -m_{k-1} & -m_{k-2} & \cdots & -m_1 & 0 \end{bmatrix}$$

$$M_1 = \begin{bmatrix} -m_k & -m_{k-1} & \cdots & -m_1 \\ -m_{k+1} & -m_k & \cdots & -m_2 \\ \vdots & \vdots & & \vdots \\ -m_{k+v-1} & -m_{k+v-2} & \cdots & -m_v \end{bmatrix}$$

The LS solution of (5.34) is obtained from

$$A^T A\mathbf{x} = A^T\mathbf{b}$$

giving the partitioned form

$$\left[\begin{array}{c|c} 2I_k & C_0 + M_0 \\ \hline C_0^T + M_0^T & C_1^T C_1 + C_0^T C_0 + M_0^T M_0 + M_1^T M_1 \end{array}\right]\begin{bmatrix}\mathbf{d}\\ \hline \mathbf{e}\end{bmatrix}$$

$$= \begin{bmatrix}\mathbf{p}\\ \hline C_1^T\mathbf{p} + M_0^T\mathbf{q} + M_1^T\mathbf{u}\end{bmatrix} \qquad (5.35)$$

which is equivalent to solving the two sets of equations

$$2\mathbf{d} + (C_0 + M_0)\mathbf{e} = \mathbf{q} \qquad (5.36)$$

and

$$(C_0^T + M_0^T)\mathbf{d} + (C_1^T C_1 + C_0^T C_0 + M_0^T M_0 + M_1^T M_1)\mathbf{e}$$

$$= C_1^T\mathbf{p} + M_0^T\mathbf{q} + M_1^T\mathbf{u} \qquad (5.37)$$

for **d** and **e**. Solving (5.36) for **d** and substituting in (5.37) gives the single vector equation

$$\left\{C_1^T C_1 + \tfrac{1}{2}(M_0 - C_0)^T(M_0 - C_0) + M_1^T M_1\right\}\mathbf{e}$$

$$C_1^T\mathbf{p} + \tfrac{1}{2}(M_0 - C_0)^T\mathbf{q} + M_1^T\mathbf{u} \qquad (5.38)$$

from which the denominator coefficient vector **e** may be calculated.

The numerator coefficient vector **d** is then obtained by substituting **e** into (5.36) to give

$$\mathbf{d} = \tfrac{1}{2}\{\mathbf{q} - (C_0 + M_0)\mathbf{e}\} \tag{5.39}$$

This is seen to be of the same general form as the upper partition component of the vector given in (5.32). It is clear that $-C_0\mathbf{e}$ is the matrix-vector representation of the expressions for $d_i$, $i = 0, 1, \ldots, k-1$, from (5.16), and $-M_0\mathbf{e}+\mathbf{q}$ is the matrix-vector representation of the expressions for the same $d_i$ from (5.33). Hence, the expression $-C_0\mathbf{e}-M_0\mathbf{e}+\mathbf{q}$ is the vector with entries $2d_i$, $i = 0, 1, \ldots, k-1$, obtained by adding the corresponding pairs of the numerator coefficient equations of (5.16) and (5.33). Equation (5.36) then indicates that all $k$ numerator coefficients $d_i$, $i = 0, 1, \ldots, k-1$, are calculated by the *average* of their values given in (5.16) and (5.33) respectively.

The corresponding partial LS method will solve the matrix equation

$$H\mathbf{e} = \mathbf{g}$$

for **e** by least-squares, where the partitioned form

$$\begin{bmatrix} C_1 \\ \hline \frac{1}{\sqrt{2}}(M_0 - C_0) \\ \hline M_1 \end{bmatrix}\mathbf{e} = \begin{bmatrix} \mathbf{p} \\ \hline \frac{1}{\sqrt{2}}\mathbf{q} \\ \hline \mathbf{u} \end{bmatrix} \tag{5.40}$$

is adopted. This is an extension of the partitioned form found in (5.18) and can be interpreted as subtracting the two middle partitions in (5.34) to eliminate **d** before multiplying the resultant equations by the factor $\frac{1}{\sqrt{2}}$. From (5.40) it is seen that the LS solution is obtained from

$$\left\{C_1^T C_1 + \tfrac{1}{2}(M_0 - C_0)^T (M_0 - C_0) + M_1^T M_1\right\}\mathbf{e}$$

$$= C_1^T \mathbf{p} + \tfrac{1}{2}(M_0 - C_0)^T \mathbf{q} + M_1^T \mathbf{u}$$

which is, as would be expected, identical to (5.38) for the full LS method.

**Relationship to other Methods**

Lucas and Munro (1991) follow up their partial LS method denominator calculation with a numerator calculation that matches time moments and/or Markov parameters exactly. In other words, certain numerator coefficient equations from (5.16) and (5.17) are simply ignored instead of 'averaging' the repeated coefficient equations as above. Further, the full LS method obtains the numerator coefficients, in part, by averaging the time moment and Markov parameter information, thus destroying the preservation of the time moments $c_i$, $i = k-r,\ k-r+1,\ \ldots\ ,\ k-1$,. Indeed, this explains Aguirre's comment (1992) regarding the need to multiply the reduced transfer function by a constant factor to *ensure* retention of the first time moment $c_0$, but notice that this is only necessary if $r \geq k$.

All of the LS techniques proposed in the literature are now seen to be special cases within the general framework given in this section, distinguished by different values chosen for $r$. Shoji *et al* (1985) and Lucas and Beat (1990) are methods equivalent to $r = 0$ (using time moments only); Lalonde *et al* (1992) use $r = k + v$ and $t = 0$ in the discrete-time case (using Markov parameters only); Lucas and Munro (1991), as described above, essentially ignore the 'averaging ' process for numerator calculation and (5.24) is solved instead of (5.30) for the denominator; finally, Aguirre's technique (1992, 1994a, 1994b) is seen to give identical results to the two-stage method of solving (5.30) and (5.32).

**Error Minimization**

The previous results help to give a deeper understanding of how the general LS Padé reduction method approximates the full system. It is now shown in what sense the LS process is used to calculate the reduced model's denominator.

First we shall consider the case of $r < k$. There has already been occasion to note (see section 5.2) that the LS solution to the linear set of equations denoted by

$$C\mathbf{x} = \mathbf{f}$$

minimizes the Euclidean norm of the vector $C\mathbf{x} - \mathbf{f}$. Consequently, the LS solution of (5.30) minimizes the Euclidean norm of $H\mathbf{e} - \mathbf{g}$, where

$$H = \begin{bmatrix} C_1 \\ \hline \frac{1}{\sqrt{2}}(M_0 - C_{0,r}) \end{bmatrix} \quad \text{and} \quad \mathbf{g} = \begin{bmatrix} \mathbf{p} \\ \hline \frac{1}{\sqrt{2}}\mathbf{q} \end{bmatrix}$$

This means that

$$\left\| \begin{array}{c} C_1\mathbf{e} - \mathbf{p} \\ \hline \frac{1}{\sqrt{2}}(M_0 - C_{0,r})\mathbf{e} - \frac{1}{\sqrt{2}}\mathbf{q} \end{array} \right\|$$

is minimized, and a closer examination of the upper and lower partition components of the vector proves to be informative.

Looking first at the upper partition components given by the vector $C_1\mathbf{e} - \mathbf{p}$, notice that, following on from the special case of LS moment matching considered in section 5.2, $\mathbf{p}$ is the vector containing the first $k + t$ time moments of $G(s)$ and $C_1\mathbf{e}$ contains the *estimates* of the corresponding reduced model's time moments, as given by the last $k + t$ of (5.16) and (5.17). $C_1\mathbf{e}$ provides only estimates because $C_1$ contains the time moments of $G(s)$ and not the reduced model $R(s)$. Likewise, examination of the lower partition components reveals that the vector $\mathbf{q}$ contains the first $r$ Markov parameters of $G(s)$ and $(M_0 - C_{0,r})\mathbf{e}$ contains the corresponding *estimates* of the

reduced model's Markov parameters. Again, because $M_0$ and $C_{0,r}$ contain the full systems's Markov parameters and time moments respectively, then $(M_0 - C_{0,r})\mathbf{e}$ gives the estimates of the first $r$ Markov parameters of $R(s)$, as calculated from substitution of equations from (5.17) into the $r$ relevant equations of (5.16).

Hence, it is now clear that the error index $J$ being minimized for the denominator calculation is given by

$$J^2 = \sum_{i=0}^{k+t-1} (c_i - c_i^*)^2 + \tfrac{1}{2}\sum_{j=1}^{r} (m_j - m_j^*)^2 \qquad (5.41)$$

where $c_i$ and $m_j$ are system time moments and Markov parameters respectively and $c_i^*$ and $m_j^*$ are their corresponding *estimates* for the reduced model. (5.41) reveals an interesting property of the LS Padé method, in that the reduced denominator is obtained by error minimization which is weighted in favour of the time moments, due to the factor of $\frac{1}{2}$ associated with the Markov parameters.

A different form for the error index is minimized for $r \geq k$. It is seen once more that the LS solution of (5.40) minimizes the Euclidean norm of $H\mathbf{e} - \mathbf{g}$ where

$$H = \begin{bmatrix} C_1 \\ \hline \frac{1}{\sqrt{2}}(M_0 - C_0) \\ \hline M_1 \end{bmatrix} \quad \text{and} \quad \mathbf{g} = \begin{bmatrix} \mathbf{p} \\ \hline \frac{1}{\sqrt{2}}\mathbf{q} \\ \hline \mathbf{u} \end{bmatrix}$$

so that

$$\left\| \begin{array}{c} C_1\mathbf{e} \\ \hline \frac{1}{\sqrt{2}}(M_0 - C_0)\mathbf{e} - \frac{1}{\sqrt{2}}\mathbf{q} \\ \hline M_1\mathbf{e} - \mathbf{u} \end{array} \right\|$$

is minimized. Again notice that $C_1\mathbf{e}$, $(M_0 - C_0)\mathbf{e}$ and $M_1\mathbf{e}$ are the *estimates* of the first $k + t$ time moments, the first $k$ Markov parameters and the next $v$ Markov parameters of the reduced model respectively. Hence, the error index minimized for the denominator calculation is given by $J$, where

$$J^2 = \sum_{i=0}^{k+t-1}\left(c_i - c_i^*\right)^2 + \frac{1}{2}\sum_{j=1}^{k}\left(m_j - m_j^*\right)^2 + \sum_{j=k+1}^{k+v}\left(m_j - m_j^*\right)^2 \qquad (5.42)$$

The $c_i$ and $m_j$ are the system time moments and Markov parameters respectively and the $c_i^*$ and $m_j^*$ are their corresponding estimates for the reduced model. It is seen that in this index the first $k + t$ time moments and the latter $v$ Markov parameters are given double the weighting of the first $k$ Markov parameters.

To sum up, the LS model reduction process minimizes the error criterion $J$, involving the first $k + t$ time moments and $r$ Markov parameters of $G(s)$ for the denominator calculation. It then calculates the numerator by matching the first $k - r$ ($r < k$) time moments of $G(s)$ to $R(s)$ followed by an *averaging* of the remaining $r$ time moment and Markov parameter equation pairs. Notice that there is no actual minimization between the system and the corresponding reduced model's time moments and Markov parameters as has been previously thought.

**Illustrative Example**

To illustrate the properties of the general LS Padé method, consider the eighth-order system with transfer function (Shamash 1975) given by

$$G(s)=\frac{40320+185760s+222088s^2+122664s^3+36382s^4+5982s^5+514s^6+18s^7}{40320+109584s+118124s^2+67284s^3+22449s^4+4536s^5+546s^6+36s^7+s^8}$$

the first four time moments and Markov parameters of which are

$$\begin{aligned} c_0 &= 1 & & m_1 = 18 \\ c_1 &= 1.8893 & \text{and} & m_2 = -134 \\ c_2 &= -2.5563 & & m_3 = 978 \\ c_3 &= 2.7863 & & m_4 = -7310 \end{aligned}$$

repectively. The general LS method is used to calculate third-order reduced models that may be divided naturally into three distinct cases so that the parameter preservation properties of the method may be clearly observed. As usual, for information, the relative impulse and step response integral square errors $I_{rel}$ and $J_{rel}$ respectively are also listed.

**Case 1**

The method is applied using $2k + t$ time moments and $r$ Markov parameters where $r < k$ with $k$=3 and $t = 2$. It is expected that the first $k$ - $r$ time moments will be preserved in calculating the $k$ - $r$ numerator coefficients, the remaining $r$ being calculated by an average of time moment and Markov parameter information.

For $r = 1$ the reduced model is

$$R_1(s)=\frac{13.5584+48.1878s+17.9993s^2}{13.5584+22.5719s+10.0128s^2+s^3}$$

with

$$I_{rel}=0.00281\% \quad \text{and} \quad J_{rel}=0.00114\%$$

and initial time moments

$$\bar{c}_0 = 1, \qquad \bar{c}_1 = 1.8893, \qquad \bar{c}_2 = -2.5572$$

It is confirmed that the first two time moments exactly match those of the full system and the numerator coefficient of $s^2$ is obtained by the averaging process.

For $r = 2$ the method gives

$$R_2(s) = \frac{15.0471 + 52.8659s + 18.0481s^2}{15.0471 + 24.4429s + 10.3811s^2 + s^3}$$

with

$$I_{rel} = 0.00321\% \quad \text{and} \quad J_{rel} = 0.00296\%$$

and initial time moments

$$\bar{c}_0 = 1, \qquad \bar{c}_1 = 1.8889, \qquad \bar{c}_2 = -2.5588$$

which exactly matches only the first time moment of the full system, and the numerator coefficients of $s$ and $s^2$ are obtained by the averaging process.

**Case 2**

Still using $2k + t$ time moments and $r$ Markov parameters, reduced models are derived for $r \geq k$ with $k = 3$ and $t = 2$. This time it is expected that no time moment preservation will be observed and the resulting models confirm this property.

For $r = 3$, the reduced model gives

$$R_3(s) = \frac{18.6374 + 64.0732s + 17.8329s^2}{18.6139 + 28.7055s + 11.0152s^2 + s^3}$$

with

$$I_{rel} = 0.01355\%$$

and the time moments

$$\bar{c}_0 = 1.0013, \qquad \bar{c}_1 = 1.9001, \qquad \bar{c}_2 = -2.5640$$

and, for $r = 4$, the reduced model gives

$$R_4(s) = \frac{23.3536 + 79.4697s + 17.3794s^2}{23.9031 + 34.9539s + 11.8236s^2 + s^3}$$

with

$$I_{rel} = 0.08180\%$$

and the time moments

$$\bar{c}_0 = 0.9770, \qquad \bar{c}_1 = 1.8623, \qquad \bar{c}_2 = -2.4908$$

The numerators of $R_3(s)$ and $R_4(s)$ are seen to be derived from the averaging process.

**Case 3**

Finally, to illustrate the same preservation properties for Markov parameters, reduced models are calculated using $t$ time moments and $2k + v$ Markov parameters with $k = 3$ and $v = 2$. The models obtained $t = 1$, 2 and 3 are given below and a comparison of their Markov parameters with those of $G(s)$ shows the preservation of the first $k$ - $t$ Markov parameters, where $t < k$, and no preservation for $t = k$. Also, the averaging process for certain numerator coefficient calculations is confirmed. When $t = 1$,

$$R_5(s) = \frac{-688.9272 - 133.9713s + 18s^2}{-383.0640 - 109.5879s + 0.0016s^2 + s^3}$$

is obtained with Markov parameters

$$\bar{m}_1 = 18, \qquad \bar{m}_2 = -134, \qquad \bar{m}_3 = 1283.9$$

For $t = 2$,

$$R_6(s) = \frac{-684.9856 - 481.2778s + 18s^2}{-380.7232 - 109.2814s + 0.0014s^2 + s^3}$$

is obtained with Markov parameters

$$\bar{m}_1 = 18, \qquad \bar{m}_2 = -481.3, \qquad \bar{m}_2 = 1282.8$$

and $t = 3$ produces

$$R_7(s) = \frac{-680.2077 - 478.4149s + 389.1139s^2}{-377.8859 - 108.9099s + 0.0011s^2 + s^3}$$

giving the Markov parameters

$$\overline{m}_1 = 389.1, \qquad \overline{m}_2 = -478.8, \qquad \overline{m}_3 = 41697.2$$

which match none of the full system's Markov parameters.

It is noticed that in cases 1 and 2 the sizes of the relative integral square errors, although small, increase as more Markov parameters are used in the general LS method. Indeed, in case 2 the values of $I_{rel}$ have increased by an order of magnitude over those recorded in case 1. This clearly reflects that the inclusion of large magnitude Markov parameters adversely affects the 'goodness of fit' of the reduced model for $r \geq k$. Further, in case 3, it can be seen that the use of Markov parameters whose magnitudes increase rapidly leads to the generation of unstable reduced models; this is a consequence of system poles being larger than one in modulus (Lucas and Munro 1991).

**Further Example**

An example due to Shoji *et al* (1985) is used to demonstrate the effectiveness of the generalised LS Padé method over other model reduction techniques in particular cases. This fourth order system is given in the frequency domain by the transfer function

$$G(s) = \frac{-18s^3 + 7s^2 + 2s + 24}{s^4 + 10s^3 + 35s^2 + 50s + 24}$$

Applying the LS Padé method matching 4 time moments and 3 Markov parameters gives the reduced model

$$R_{LS}(s) = \frac{-18s^2 - 1.208s + 19.864}{s^3 + 10.456s + 38.745s + 19.864}$$

with relative ISE results

$$I_{rel} = 2.8\% \qquad \text{and} \qquad J_{rel} = 7.3\%$$

It is noted that, in this example, the exact Padé method does not produce a result due to numerical difficulties and LS moment matching fails to produce a stable third order model.

Its superiority over other methods is made clear by the relative ISE figures for the third order models produced by the stability preserving methods. Pole retention gives the transfer function

$$R_{PR}(s) = \frac{2s^2 - s + 6}{s^3 + 6s^2 + 11s + 6}$$

with

$$I_{rel} = 121\% \qquad \text{and} \qquad J_{rel} = 42\%$$

Routh approximation produces the third order model given by

$$R_{RA}(s) = \frac{1.874s^2 + 0.2389s + 2.8761}{s^3 + 4.0992s^2 + 5.992s + 2.8761}$$

where

$$I_{rel} = 117\% \qquad \text{and} \qquad J_{rel} = 39\%$$

Finally, the Stability Equation method provided the third order result

$$R_{SE} = \frac{7s^2 + 2s + 24}{10s^3 + 34.3s^2 + 50s + 24}$$

with errors of

**Fig 5.5 Further Example: Third Order Models**

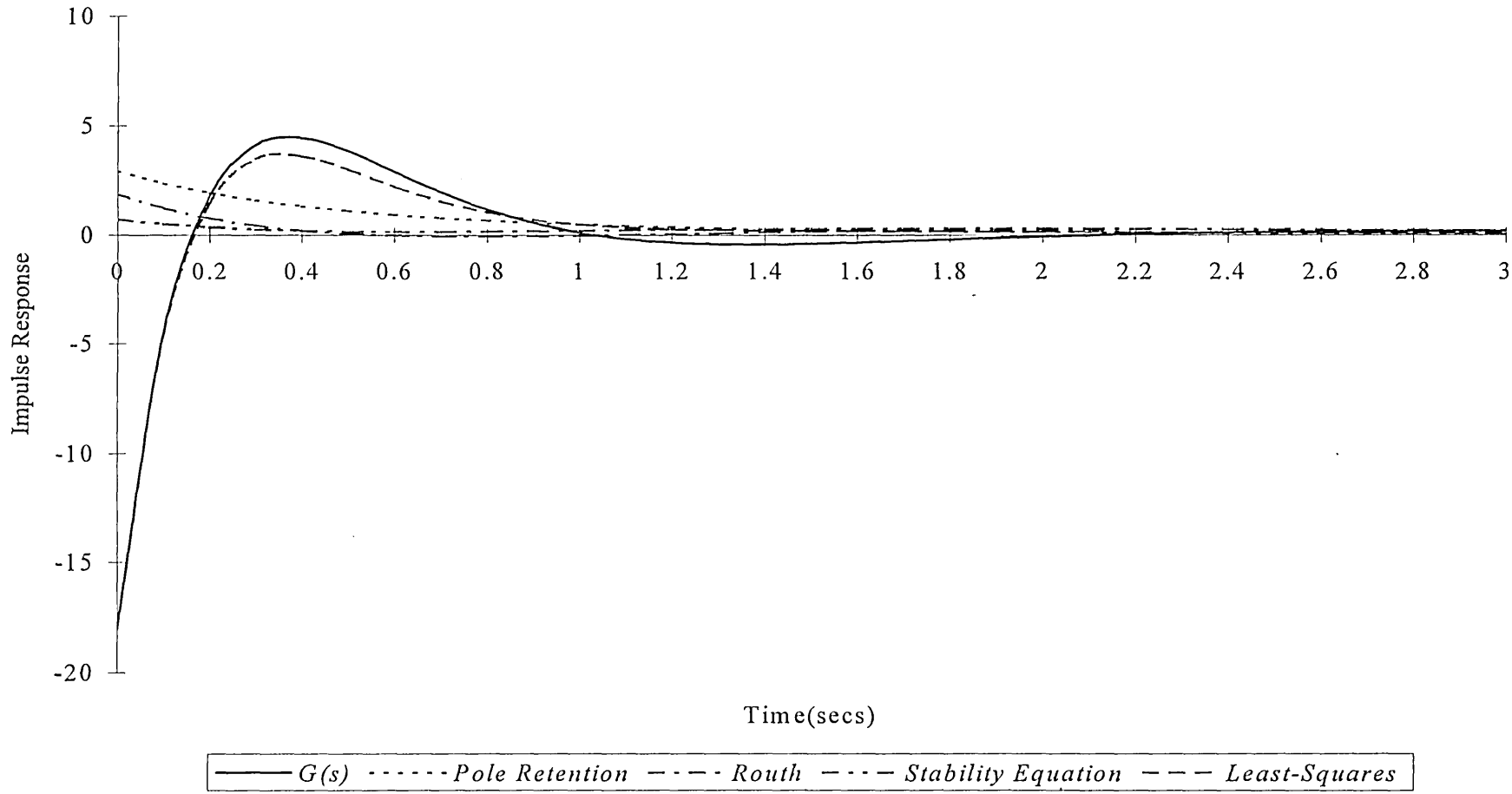
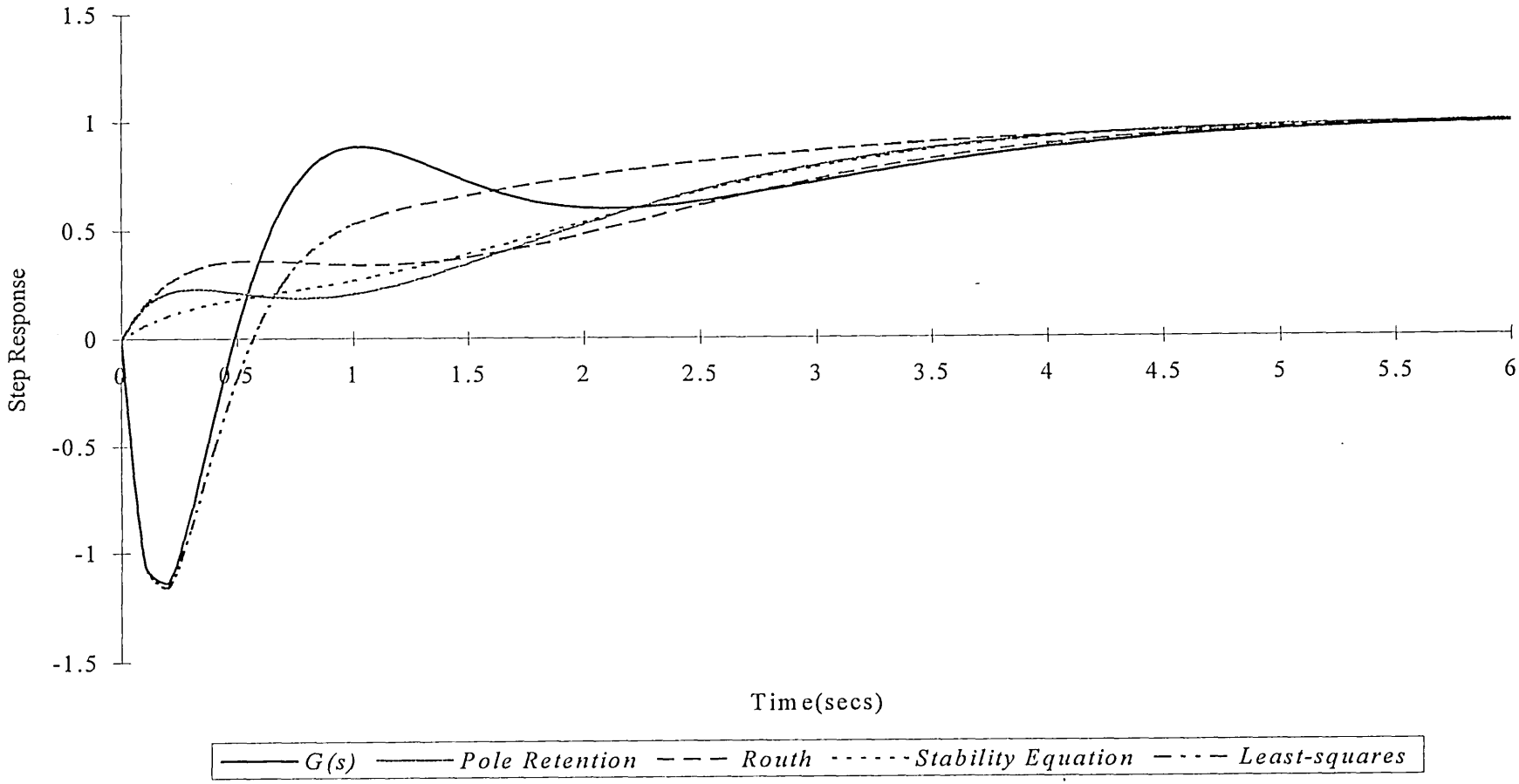


**Fig 5.6 Further Example: Third Order Models**



$$I_{rel} = 104\% \qquad \text{and} \qquad J_{rel} = 35\%$$

In all three cases the relative ISE figures are much larger and far less acceptable than those associated with $R_{LS}(s)$ above. Comparison graphs are given for impulse and step responses in figures 5.5 and 5.6 respectively.

## 5.6 Weighted LS Padé Approximation

In the last section the observation was made that the error index given by (5.41) involves a weighting in favour of the time moments by a factor of two. This suggests that it might be useful in some examples to use different weighting factors for the time moment and Markov parameter information respectively. The usefulness of weighting techniques has been considered in the literature by Davidson and Walters (1988).

The present section will apply the analysis used earlier in this chapter to the general case where arbitrary weights are applied to the system parameters used in the LS approximation. First, a single weighting factor applied to all Markov parameters is considered. Then a further single arbitrary weighting factor is introduced for the time moment information. Finally, consideration is given to the more general case where $2k + t + r$ distinct weightings are applied to the $2k + t$ time moments and $r$ Markov parameters used for the LS approximation.

### Weight $w_m$ applied to Markov parameters

In the case of the full LS Padé method with weight $w_m$ applied to the $r$ Markov parameters, both sides of the equation (5.18)

$$A\mathbf{x} = \mathbf{b}$$

are multiplied by a weighting matrix

$$W = \begin{bmatrix} I_{k+l} & \Phi & \Phi \\ \Phi & I_k & \Phi \\ \Phi & \Phi & w_m I_r \end{bmatrix}$$

giving

$$WA\mathbf{x} = W\mathbf{b}$$

which in partitioned form becomes

$$\left[\begin{array}{c|c} \Phi & C_1 \\ \hline I_k & C_0 \\ \hline w_m\lambda & w_m M_0 \end{array}\right]\begin{bmatrix}\mathbf{d}\\ \mathbf{e}\end{bmatrix} = \begin{bmatrix}\mathbf{p}\\ \mathbf{0}\\ w_m\mathbf{q}\end{bmatrix}$$

Since the transpose of the pre-multiplying matrix is

$$(WA)^T = \left[\begin{array}{c|c|c} \Phi & I_k & w_m\lambda^T \\ \hline C_1^T & C_0^T & w_m M_0^T \end{array}\right]$$

the vector **x** is given by the solution of

$$\left[\begin{array}{c|c} I_k + w_m^2\lambda^T\lambda & C_0 + w_m^2\lambda^T M_0 \\ \hline C_0^T + w_m^2 M_0^T\lambda & C_1^T C_1 + C_0^T C_0 + w_m^2 M_0^T M_0 \end{array}\right]\begin{bmatrix}\mathbf{d}\\ \mathbf{e}\end{bmatrix} = \left[\begin{array}{c} w_m^2\lambda^T\mathbf{q} \\ \hline C_1^T\mathbf{p} + w_m^2 M_0^T\mathbf{q} \end{array}\right]$$

Therefore the solution for the full LS Padé method with Markov parameters weighted using a constant factor $w_m$ will be given by the system of equations

$$\left(I_k + w_m^2\lambda^T\lambda\right)\mathbf{d} + \left(C_0 + m_m^2\lambda^T M_0\right)\mathbf{e} = w_m^2\lambda^T\mathbf{q} \qquad (5.43)$$

and

$$\left(C_0^T + w_m^2 M_0^T\lambda\right)\mathbf{d} + \left(C_1^T C_1 + C_0^T C_0 + w_m^2 M_0^T M_0\right)\mathbf{e} = C_1^T\mathbf{p} + w_m^2 M_0^T\mathbf{q} \qquad (5.44)$$

Substituting the solution of (5.43) for **d** in equation (5.44)gives an equation for the denominator coefficient vector **e**

$$\left\{-\left(C_0+w_m^2\lambda^T M_0\right)^T\left(I_k+w_m^2\lambda^T\lambda\right)^{-1}\left(C_0+w_m^2\lambda^T M_0\right)+C_1^T C_1+C_0^T C_0+w_m^2 M_0^T M_0\right\}\mathbf{e}$$

$$=C_1^T\mathbf{p}+w_m^2 M_0^T\mathbf{q}-\left(C_0+w_m^2\lambda^T M_0\right)^T\left(I_k+w_m^2\lambda^T\lambda\right)^{-1}w_m^2\lambda^T\mathbf{q} \qquad (5.45)$$

It is of interest to compare this equation with (5.23). We can see that they are identical apart from the factor of $w_m{}^2$ multiplying the $\lambda$ and $M_0$ terms in (5.45).

Using the definitions of $C_{0,r}$ and $\lambda$ given in section 5.5 (subsection entitled Partial Least-Squares Method), it is noted that the following identities hold

$$C_0^T C_0 = C_{0,r}^T C_{0,r} + C_{0,k-r}^T C_{0,k-r}$$

$$w_m^2\lambda^T\lambda=\left[\begin{array}{c|c} w_m^2 I_r & \Phi \\ \hline \Phi & \Phi \end{array}\right]$$

$$C_0+w_m^2\lambda^T M_0=\left[\begin{array}{c} C_{0,r}+w_m^2 M_0 \\ \hline C_{0,k-r} \end{array}\right] \qquad (5.46)$$

$$I_k+w_m^2\lambda^T\lambda=\left[\begin{array}{c|c} \left(1+w_m^2\right)I_r & \Phi \\ \hline \Phi & I_{k-r} \end{array}\right]$$

where $C_{0,k-r}$ is the $(k-r)\times k$ matrix consisting of the *last* $k$ - $r$ rows of $C_0$ and the $\Phi$'s are null matrices of appropriate dimension. It is further noticed that since $I_k+w_m^2\lambda^T\lambda$ is a diagonal matrix this gives

$$\left(I_k+w_m^2\lambda^T\lambda\right)^{-1}=\left[\begin{array}{c|c} \dfrac{1}{\left(1+w_m^2\right)}I_r & \Phi \\ \hline \Phi & I_{k-r} \end{array}\right] \qquad (5.47)$$

so that substituting the identities given in (5.46) and (5.47) gives (5.45) as

$$\left(C_1^T C_1+\frac{w_m^2}{1+w_m^2}C_{0,r}^T C_{0,r}-\frac{w_m^2}{1+w_m^2}C_{0,r}^T M_0-\frac{w_m^2}{1+w_m^2}M_0^T C_{0,r}+\frac{w_m^2}{1+w_m^2}M_0^T M_0\right)\mathbf{e}$$

$$=C_1^T\mathbf{p}+\frac{w_m^2}{1+w_m^2}M_0^T\mathbf{q}-\frac{w_m^2}{1+w_m^2}C_{0,r}^T\mathbf{q} \qquad (5.48)$$

which can be written in factorised form

$$\left\{C_1^T C_1 + \frac{w_m^2}{1+w_m^2}\left(M_0 - C_{0,r}\right)^T\left(M_0 - C_{0,r}\right)\right\}\mathbf{e} = C_1^T\mathbf{p} + \frac{w_m^2}{1+w_m^2}\left(M_0 - C_{0,r}\right)^T\mathbf{q} \quad (5.49)$$

This equation is of the same form as (5.29) with the factor of ½ in (5.29) replaced by the factor

$$\frac{w_m^2}{1+w_m^2}$$

Hence, the equivalence between the full and partial LS methods observed in section 5.5 holds also in this case if we adopt the modified partitioned form for $H$ and $\mathbf{g}$ given by

$$\left[\begin{array}{c} C_1 \\ \hline \dfrac{w_m}{\sqrt{1+w_m^2}}\left(M_0 - C_{0,r}\right) \end{array}\right]\mathbf{e} = \left[\begin{array}{c} \mathbf{p} \\ \hline \dfrac{w_m}{\sqrt{1+w_m^2}}\mathbf{q} \end{array}\right] \quad (5.50)$$

The full LS Padé method with the Markov parameters weighted by a factor of $w_m$ is seen, therefore, to be equivalent to calculating the LS solution of (5.50) followed by the derivation of the numerator coefficients from

$$\mathbf{d} = \left[\begin{array}{c} \dfrac{w_m^2}{1+w_m^2}\left\{\mathbf{q} - \left(C_{0,r} + M_0\right)\mathbf{e}\right\} \\ \hline -C_{0,k-r}\mathbf{e} \end{array}\right]$$

the equivalent of (5.32) in this case with the error index being minimised given by

$$J^2 = \sum_{i=0}^{k+t-1}\left(c_i - c_i^*\right)^2 + \frac{w_m^2}{1+w_m^2}\sum_{j=1}^{r}\left(m_j - m_j^*\right)^2 \quad (5.51)$$

$$\left\{ \text{For the case } r \geq k \quad J^2 = \sum_{i=0}^{k+t-1}\left(c_i - c_i^*\right)^2 + \frac{w_m^2}{1+w_m^2}\sum_{j=1}^{k}\left(m_j - m_j^*\right)^2 + w_m^2\sum_{j=k+1}^{k+v}\left(m_j - m_j^*\right)^2 \right\}$$

**Weights $w_c$ and $w_m$ applied to time moments and Markov parameters respectively**

In the case where an arbitrary weighting of $w_c$ is applied also to the $2k + t$ time moments, both sides of the equation (5.18)

$$A\mathbf{x} = \mathbf{b}$$

will be pre-multiplied by the weighting matrix

$$W = \begin{bmatrix} w_c I_{k+t} & \Phi & \Phi \\ \Phi & w_c I_k & \Phi \\ \Phi & \Phi & w_m I_r \end{bmatrix}$$

giving **x** from the solution of

$$(WA)^T WA\mathbf{x} = (WA)^T W\mathbf{b}$$

which has partitioned form

$$\left[\begin{array}{c|c} w_c^2 I_k + w_m^2 \lambda^T \lambda & w_c^2 C_0 + w_m^2 \lambda^T M_0 \\ \hline w_c^2 C_0^T + w_m^2 M_0^T \lambda & w_c^2 C_1^T C_1 + w_c^2 C_0^T C_0 + w_m^2 M_0^T M_0 \end{array}\right]\left[\begin{array}{c} \mathbf{d} \\ \hline \mathbf{e} \end{array}\right] = \left[\begin{array}{c} w_m^2 \lambda^T \mathbf{q} \\ \hline w_c^2 C_1^T \mathbf{p} + w_m^2 M_0^T \mathbf{q} \end{array}\right]$$

The usual method of substitution for **d** applied to the resulting system gives an equation very similar to (5.45) for the solution of **e**

$$\left\{-\left(w_c^2 C_0 + w_m^2 \lambda^T M_0\right)^T \left(w_c^2 I_k + w_m^2 \lambda^T \lambda\right)^{-1} \left(w_c^2 C_0 + w_m^2 \lambda^T M_0\right) + w_c^2 C_1^T C_1 + w_c^2 C_0^T C_0 + w_m^2 M_0^T M_0\right\}\mathbf{e}$$

$$= w_c^2 C_1^T \mathbf{p} + w_m^2 M_0^T \mathbf{q} - \left(w_c^2 C_0 + w_m^2 \lambda^T M_0\right)^T \left(w_c^2 I_k + w_m^2 \lambda^T \lambda\right)^{-1} w_m^2 \lambda^T \mathbf{q} \qquad (5.52)$$

This is the equation for the full weighted LS Padé method for the case under consideration. It may be represented as an equivalent partial method as in previous sections by substituting in (5.52) for the identities

$$w_c^2 C_0^T C_0 = w_c^2 C_{0,r}^T C_{0,r} + w_c^2 C_{0,k-r}^T C_{0,k-r}$$

$$w_m^2 \lambda^T \lambda = \left[\begin{array}{c|c} w_m^2 I_r & \Phi \\ \hline \Phi & \Phi \end{array}\right] \qquad w_c^2 C_0 + w_m^2 \lambda^T M_0 = \left[\begin{array}{c} w_c^2 C_{0,r} + w_m^2 M_0 \\ \hline w_c^2 C_{0,k-r} \end{array}\right] \tag{5.53}$$

$$(w_c^2 I_k + w_m^2 \lambda^T \lambda)^{-1} = \left[\begin{array}{c|c} \dfrac{1}{w_c^2 + w_m^2} I_r & \Phi \\ \hline \Phi & \dfrac{1}{w_c^2} I_{k-r} \end{array}\right]$$

to obtain the simplified and factorised form

$$\left\{ w_c^2 C_1^T C_1 + \frac{w_c^2 w_m^2}{w_c^2 + w_m^2} (M_0 - C_{0,r})^T (M_0 - C_{0,r}) \right\} \mathbf{e} = w_c^2 C_1^T \mathbf{p} + \frac{w_c^2 w_m^2}{w_c^2 + w_m^2} (M_0 - C_{0,r})^T \mathbf{q}$$

In this case the equivalent method is achieved by giving (5.24) the form

$$\left[\begin{array}{c} w_c C_1 \\ \hline \dfrac{w_c w_m}{\sqrt{w_c^2 + w_m^2}} (M_0 - C_{0,r}) \end{array}\right] \mathbf{e} = \left[\begin{array}{c} w_c \mathbf{p} \\ \hline \dfrac{w_c w_m}{\sqrt{w_c^2 + w_m^2}} \mathbf{q} \end{array}\right] \tag{5.54}$$

when the full method is seen to be equivalent to solving (5.54) for **e** and then calculating the numerator from

$$\mathbf{d} = \left[\begin{array}{c} \dfrac{w_c^2 w_m^2}{w_c^2 + w_m^2} \{\mathbf{q} - (C_{0,r} + M_0)\mathbf{e}\} \\ \hline - w_c^2 C_{0,k-r} \mathbf{e} \end{array}\right]$$

Clearly, equation (5.32) is simply the special form of this solution for **d** with the weights, $w_c$ and $w_m$, equal to unity and the error index minimised is given by

$$J^2 = w_c^2 \sum_{i=0}^{k+t-1} (c_i - c_i^*)^2 + \frac{w_c^2 w_m^2}{w_c^2 + w_m^2} \sum_{j=1}^{r} (m_j - m_j^*)^2 \tag{5.55}$$

$$\left\{ \text{For the case } r \geq k \quad J^2 = w_c^2 \sum_{i=0}^{k+t-1} (c_i - c_i^*)^2 + \frac{w_c^2 w_m^2}{w_c^2 + w_m^2} \sum_{j=1}^{k} (m_j - m_j^*)^2 + w_m^2 \sum_{j=k+1}^{k+v} (m_j - m_j^*)^2 \right\}$$

**Arbitrary and distinct weights $w_i$ applied to all system parameters**

The most general case of applying $2k + t + r$ distinct weights to the time moments and Markov parameters is suggested by the work of Davidson and Walters (1988). However, some observations will serve to demonstrate that this case is inaccessible to the two-stage analysis which has been applied to previous, more restricted cases.

Observation of the equations (5.28), (5.48) and (5.49) for the partial methods discussed up to this point indicate that the equivalent two-stage analysis is dependent upon being able to extract numerical factors resulting in the partitioned form of (5.32). Consider first the case of the generalised LS Padé method analysed in section 5.5. The numerical factor of ½ is seen to enable the equivalent full and partial methods, coming from the $r \times r$ matrix

$$\begin{bmatrix} ½ & 0 & \cdots & 0 \\ 0 & ½ & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & ½ \end{bmatrix} = \tfrac{1}{2} I_r$$

which appears as a partition in (5.27). Similarly, in the more general case of weights $w_c$ and $w_m$ applied to time moments and Markov parameters respectively (the previous subsection), the numerical factors are seen to be

$$\frac{1}{w_c^2} \qquad \text{and} \qquad \frac{1}{w_c^2 + w_m^2}$$

which may be removed from the matrices forming the partitions of

$$(w_c^2 I_k + w_m^2 \lambda^T \lambda)^{-1} = \left[\begin{array}{c|c} \frac{1}{w_c^2 + w_m^2} I_r & \Phi \\ \hline \Phi & \frac{1}{w_c^2} I_{k-r} \end{array}\right]$$

In other words, in all the cases considered up to this point the diagonal matrix whose inverse is required for the two-stage analysis of the full LS method has all leading block diagonal elements equal. Indeed, if the elements in all leading block diagonals had not all been equal then it would not have been possible to express the partitions of the inverse as the product of a constant scalar factor and the identity matrix of appropriate dimension.

However, in the case of distinct weights $w_i$ ($i = 1, 2, \ldots, 2k + t + r$) the weighting matrix is

$$W = \begin{bmatrix} \Gamma & \Phi & \Phi \\ \Phi & \Delta & \Phi \\ \Phi & \Phi & \mathrm{K} \end{bmatrix}$$

where

$$\Gamma = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{k+t} \end{bmatrix} \qquad \Delta = \begin{bmatrix} w_{k+t+1} & 0 & \cdots & 0 \\ 0 & w_{k+t+2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{2k+t} \end{bmatrix}$$

$$\mathrm{K} = \begin{bmatrix} w_{2k+t+1} & 0 & \cdots & 0 \\ 0 & w_{2k+t+2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{2k+t+r} \end{bmatrix}$$

and the diagonal matrix whose inverse would be required for the two-stage analysis is

$$\Delta^2 + \lambda^T \mathrm{K}^2 \lambda = \left[\begin{array}{c|c} \Delta_1^2 + \mathrm{K}^2 & \Phi \\ \hline \Phi & \Delta_2^2 \end{array}\right]$$

$$\Delta_1 = \begin{bmatrix} w_{k+t+1} & 0 & \cdots & 0 \\ 0 & w_{k+t+2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{k+t+r} \end{bmatrix} \quad \Delta_2 = \begin{bmatrix} w_{k+t+r+1} & 0 & \cdots & 0 \\ 0 & w_{k+t+r+2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{2k+t} \end{bmatrix}$$

where

Clearly, since the elements in the leading diagonals are not all equal, neither this diagonal matrix nor its inverse can be expressed in the form

$$\left[\begin{array}{c|c} f_1 I_r & \Phi \\ \hline \Phi & f_2 I_{k-r} \end{array}\right]$$

where $f_1$ and $f_2$ are scalar factors. Therefore, no such factor can be identified in this case to establish a partial method equivalent to the full method.

# CHAPTER 6

# EXTENSION OF THE FRAMEWORK TO DISCRETE-TIME SYSTEMS

## 6.1 Introduction

In section 4.4, the idea of LS Padé approximation using Markov parameters only in the discrete-time case was introduced. In this section, the equivalence of the full and partial LS methods is shown to hold also for discrete-time systems. Further, it is shown that the LS Padé method applied to discrete-time systems possesses a stability preserving property (Lucas and Smith 1998) that enhances its use considerably and that the nonuniqueness property shown in section 5.3 for the continuous-time case is shown to be important in relation to the reduced model's stability.

**Equivalence of the Full and Partial Methods**

The system of equations (4.15) given in section 4.6 may be written in matrix-vector form as

$$A\mathbf{x} = \mathbf{b}$$

or

$$\left[\begin{array}{c|c} I_{k+1} & M \\ \hline \Phi & H \end{array}\right]\left[\begin{array}{c} \mathbf{d} \\ \hline \mathbf{e} \end{array}\right] = \left[\begin{array}{c} \mathbf{p} \\ \hline \mathbf{q} \end{array}\right] \tag{6.1}$$

where $I_{k+1}$ is the $(k+1)\times(k+1)$ unit matrix, $\Phi$ is the $(k+r)\times(k+1)$ null matrix, $H$, $\mathbf{q}$ and $\mathbf{e}$ are as defined in (4.16), and

$$\mathbf{d} = [d_k \quad d_{k-1} \quad \ldots \quad d_0]^T$$

$$\mathbf{p} = [m_0 \quad m_1 \quad \ldots \quad m_k]^T$$

$$\text{and} \quad M = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ -m_0 & 0 & 0 & \cdots & 0 \\ -m_1 & -m_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -m_{k-1} & -m_{k-2} & \cdots & -m_0 & 0 \end{bmatrix}$$

To show that the LS solution of (6.1) yields the same as that of (4.16) along with the first $k+1$ equations from (4.14), notice that

$$A^T A = \left[\begin{array}{c|c} I_{k+1} & M \\ \hline M^T & M^T M + H^T H \end{array}\right]$$

and

$$A^T \mathbf{b} = \left[\begin{array}{c} \mathbf{p} \\ \hline M^T \mathbf{p} + H^T \mathbf{q} \end{array}\right]$$

So the LS solution is given by

$$\left[\begin{array}{c|c} I_{k+1} & M \\ \hline M^T & M^T M + H^T H \end{array}\right] \left[\begin{array}{c} \mathbf{d} \\ \hline \mathbf{e} \end{array}\right] = \left[\begin{array}{c} \mathbf{p} \\ \hline M^T \mathbf{p} + H^T \mathbf{q} \end{array}\right]$$

that is

$$\mathbf{d} + M\mathbf{e} = \mathbf{p} \tag{6.2}$$

and

$$M^T \mathbf{d} + \left(M^T M + H^T H\right)\mathbf{e} = M^T \mathbf{p} + H^T \mathbf{q} \tag{6.3}$$

The similarity to (5.9) and (5.10) in the continuous-time case is obvious and eliminating $\mathbf{d}$ from (6.2) and (6.3) gives

$$H^T H\mathbf{e} = H^T \mathbf{q} \tag{6.4}$$

The solution of this equation is identical to that given by (4.16) for the partial method when only the reduced denominator is found by the LS method. Further, (6.2) yields the numerator coefficients $\mathbf{d}$, once $\mathbf{e}$ has been found, which is equivalent to using the first $k+1$ equations from (4.14). Hence the equivalence of the full and partial approaches in the discrete-time case using Markov parameters only.

Clearly, the two-stage analysis into LS denominator calculation followed by numerator calculation by exact parameter matching first seen in section 5.2 may be applied also in this case. Note that in solving (6.4) the Euclidean norm of the vector $H\mathbf{e}-\mathbf{q}$ is minimised where $\mathbf{q}$ contains the last $k+r$ of the $2k+r+1$ Markov parameters of the full system being used and the $H\mathbf{e}$ contains *estimates* of the reduced model's Markov parameters. The error index being minimised is $J$, where

$$J^2 = \sum_{i=k+1}^{2k+r} \left(m_i - m_i^*\right)^2$$

the $m_i^*$ being only estimates of the reduced model's Markov parameters given by $H\mathbf{e}$.

For this reason the method does not produce optimal reduced models for least sum of square errors between time responses. Nevertheless it might be expected that the models will be "near optimal" in that they would produce these square error sums close to the optimal ones. This has been found to be the case over a range of different systems and example 6.2 in the next section illustrates this point.

## 6.2 A Stability Preservation Property

One of the main reasons for developing LS Padé methods is that they are a natural extension of the exact Padé method to possibly overcome any stability problems associated with the reduced models. Until now this appeared to be a 'hit-or-miss' way of obtaining a stable model – a feature of most Padé methods. However, for discrete-time systems Lalonde (1992a) observed an interesting phenomenon in that by letting $r\rightarrow\infty$ (the extra number of Markov parameters used for an LS approximation) a stable model tended to result for any order of reduced model. He also tried to prove this property by likening it to one of linear system identification and used obscure statistical arguments to arrive at the required result.

A mathematical proof of this stability preserving property for discrete-time systems is now given which puts the technique on a firmer theoretical foundation. Consequently, this property enhances its obvious appeal in control system design as a model reduction method.

**Proof of Stability Property**

The reduced $k$th order model's denominator polynomial is obtained by solving the equation, $H\mathbf{e} = \mathbf{q}$, in a least-squares sense as given by (6.4). This is equivalent to finding LS estimates of the characteristic polynomial contained in the difference equation

$$m_{i+k} + m_{i+k-1}e_{k-1} + \ldots + m_{i+1}e_1 + m_i e_0 = \varepsilon_i \qquad (6.5)$$

for $i = 1, 2, 3, \ldots$, where $\varepsilon_i$ is the error term in approximating $-m_{i+k}$ by

$$m_{i+k-1}e_{k-1} + \ldots + m_{i+1}e_1 + m_i e_0$$

and an infinite number of Markov parameters are used in the approximation. Notice that for a stable full system, $m_i \to 0$ as $i \to \infty$.

Using the forward-shift operator $Q$, defined by $Q^n m_i \equiv m_{i+n}$, gives (6.5) as

$$\left(Q^k + e_{k-1}Q^{k-1} + \ldots + e_1 Q + e_0\right)m_i = \varepsilon_i$$

or equivalently,

$$(Q - p_1)(Q - p_2)\ldots(Q - p_k)m_i = \varepsilon_i \qquad (6.6)$$

where $p_j$ ($j = 1, 2, 3, \ldots, k$) are the poles of the reduced order model.

Now let

$$u_i \equiv (Q - p_1)(Q - p_2)\ldots(Q - p_{j-1})(Q - p_{j+1})\ldots(Q - p_k)m_i \qquad (6.7)$$

where $j$ is arbitrarily chosen to be an integer between 1 and $k$. Clearly, (6.7) gives $u_i \to 0$ as $i \to \infty$ since $m_i \to 0$ as $i \to \infty$, i.e. the $u_i$ ($i = 1, 2, 3, \ldots$) also form a stable sequence, possibly complex. Substituting (6.7) into (6.6) gives the equivalent first order difference equation

$$(Q - p_j)u_i = \varepsilon_i$$

or

$$u_{i+1} - p_j u_i = \varepsilon_i \qquad (6.8)$$

($i = 1, 2, 3, \ldots$).

It is shown in Appendix 1 that the value of $p_j$ which minimises the sum of the squares of the errors

$$E = \sum_{i=1}^{\infty} \left| \varepsilon_i^2 \right|$$

is given by

$$p_j = \frac{\sum_{i=1}^{\infty} \hat{u}_i u_{i+1}}{\sum_{i=1}^{\infty} \hat{u}_i u_i} \qquad (6.9)$$

where $\hat{u}_i$ is the complex conjugate of $u_i$. Further notice that

$$\begin{aligned} |p_j| &= \frac{\left| \sum_{i=1}^{\infty} \hat{u}_i u_{i+1} \right|}{\sum_{i=1}^{\infty} |u_i|^2} \\ &\leq \frac{\sum_{i=1}^{\infty} |u_i||u_{i+1}|}{\sum_{i=1}^{\infty} |u_i|^2} \\ &< 1 \end{aligned}$$

for a stable sequence $u_i$; the last inequality step is proved in Appendix 2.

Allowing $j$ to take consecutively the values 1, 2, 3, ... , $k$ shows that all the reduced model's poles are less than one in modulus, hence the stability preservation property is proved.

Of course, in practice a finite number of Markov parameters would be used to derive the LS Padé model. For a stable system it is an easy task to find a suitable value of $r$ such that

$$\left| \sum_{i=1}^{\infty} m_i^2 - \sum_{i=1}^{2k+r} m_i^2 \right| \leq \delta$$

where $\delta$ is some acceptable tolerance value. For small enough $\delta$ the stability property will always hold.

**Restriction of Stability Property**

In section 5.3 it was shown for continuous-time systems that it is possible to produce different reduced $k$th order transfer functions by the LS Padé method. This is done by choosing in turn a different coefficient in the denominator polynomial to equal one and then estimating the rest by LS approximation. The same idea may be applied in the discrete-time case, but it is now shown that the stability preservation property proved above no longer holds if any coefficient, other than that of $z^k$, is chosen to equal one. This will be shown for a general second order reduced model and extension to higher orders is obvious.

For a second order model with $e_0 = 1$ equation (6.5) becomes

$$m_{i+2}e_2 + m_{i+1}e_1 + m_i = \varepsilon_i \qquad (6.10)$$

$$(e_2Q^2 + e_1Q + 1)m_i = \varepsilon_i$$

($i = 1, 2, 3, \ldots$) which factorises to

$$(\alpha Q - 1)(\beta Q - 1)m_i = \varepsilon_i \tag{6.11}$$

where $1/\alpha$ and $1/\beta$ are the poles of the reduced model.

Making the substitution, $u_i = (\beta Q - 1)m_i$, gives (6.11) as

$$(\alpha Q - 1)u_i = \varepsilon_i$$

or

$$\alpha u_{i+1} - u_i = \varepsilon_i \tag{6.12}$$

and the LS solution of (6.12) which minimises $E$ is seen to be

$$\alpha = \frac{\sum_{i=1}^{\infty} u_i \hat{u}_{i+1}}{\sum_{i=1}^{\infty} u_{i+1} \hat{u}_{i+1}}$$

giving,

$$|\alpha| = \frac{\left|\sum_{i=1}^{\infty} u_i \hat{u}_{i+1}\right|}{\sum_{i=1}^{\infty} |u_{i+1}|^2} \tag{6.13}$$

In general it is not possible to say whether $|\alpha| \leq 1$ or $|\alpha| > 1$ from (6.13) because the term $|u_1|^2$ is 'missing' from the denominator summation term, so the result in

Appendix 2 cannot be used to advantage. The same argument applies to $|\beta|$ and hence stability cannot be *guaranteed* in this case.

If, instead, $e_1 = 1$ is chosen then (6.10) becomes

$$m_{i+2}e_2 + m_{i+1} + m_i e_0 = \varepsilon_i$$

$$\left(e_2 Q^2 + Q + e_0\right)m_i = \varepsilon_i$$

($i = 1, 2, 3, \ldots$) which may be factorised into

$$(\alpha Q - 1)[\beta Q + (1+\beta)/\alpha]m_i = \varepsilon_i \qquad (6.14)$$

where $1/\alpha$ and $-(1 + \beta)/\alpha\beta$ are now the poles of the reduced model. Using the substitution, $u_i = [\beta Q + (1 + \beta)/\alpha]m_i$, in equation (6.14) leads to the same result for $|\alpha|$ as given in (6.13), showing that stability cannot be guaranteed in this case also.

Similar arguments may be extended to reduced order models of any order so that only in the case where $e_k = 1$ is stability *guaranteed.* It is interesting that this property happens to coincide with the "natural" choice of using $e_k = 1$ in LS reduced models to date.

**Example 6.1**

As a simple illustration of the stability restriction property, consider the sequence of Markov parameters from a stable system given by

$$\{m_i\} = \{6, 5, 2, 1, 0, -0.5, -0.2, 0.1, 0.08, 0.02, 0.006, 0, 0, 0, \ldots\}$$

for $i = 0, 1, 2, 3, \ldots$. Using the LS method with $r \to \infty$ to derive second order reduced models yields the following pole locations:

(i) $e_2 = 1$ gives complex poles at $0.33 \pm 0.09i$

(ii) $e_1 = 1$ gives poles at 1.64 and 0.41

(iii) $e_0 = 1$ gives poles at 3.35 and 0.45.

Notice that only when $e_2 = 1$ is the model stable.

**Example 6.2**

To illustrate the LS Padé method applied to discrete-time systems, consider the fourth order transfer function (Lucas 1993d)

$$G(z) = \frac{z^3 - 0.1z^2 - 0.47z - 0.225}{z^4 - 1.2z^3 + 0.55z^2 + 0.05z - 0.075}$$

with poles at 0.5, -0.3, $0.5 \pm 0.5i$ and zeros at 0.9, $-0.4 \pm 0.3i$. Reduction of this system to $k$th order models ($k = 3, 2$), defined by

$$R(z) = \frac{d_k z^k + d_{k-1} z^{k-1} + \cdots + d_1 z + d_0}{z^k + e_{k-1} z^{k-1} + \cdots + e_1 z + e_0}$$

for various values of $r$ (the extra number of Markov parameters used) are shown in Tables 6.1 and 6.2 respectively. Also displayed in these tables are the square error sum (SES) and *relative* square error sum ($SES_{rel}$) values, defined by

$$SES = \sum_{i=0}^{\infty} (y_i - \bar{y}_i)^2 \qquad \text{and} \qquad SES_{rel} = \frac{\sum_{i=0}^{\infty} (y_i - \bar{y}_i)^2}{\sum_{i=0}^{\infty} y_i^2}$$

where the $y_i$ and $\bar{y}_i$ are the respective full and reduced models' $i$th pulse response values.

**Table 6.1**

| $r$ | $d_2$ | $d_1$ | $d_0$ | $e_2$ | $e_1$ | $e_0$ | SES | $SES_{rel}$% |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | -2.4251 | -0.5014 | -3.5251 | 3.0762 | -1.8063 | unstable | - |
| 2 | 1 | -0.6538 | -0.2983 | -1.7538 | 1.3308 | -0.4477 | 0.0312 | 0.872 |
| 4 | 1 | -0.6031 | -0.3031 | -1.7031 | 1.2703 | -0.4023 | 0.0141 | 0.394 |
| 6 | 1 | -0.6032 | -0.3040 | -1.7032 | 1.2696 | -0.4020 | 0.0139 | 0.387 |
| 8 | 1 | -0.6021 | -0.3042 | -1.7021 | 1.2681 | -0.4009 | 0.0136 | 0.379 |
| 20 | 1 | -0.6021 | -0.3042 | -1.7021 | 1.2680 | -0.4009 | 0.0135 | 0.378 |

**Table 6.2**

| $r$ | $d_1$ | $d_0$ | $e_1$ | $e_0$ | SES | $SES_{rel}$% |
|---|---|---|---|---|---|---|
| 0 | 1 | 0.1659 | -0.9341 | 0.7275 | 0.7083 | 19.787 |
| 2 | 1 | 0.0304 | -1.0696 | 0.7769 | 0.6613 | 18.474 |
| 4 | 1 | 0.0383 | -1.0617 | 0.7020 | 0.2817 | 7.868 |
| 6 | 1 | 0.0473 | -1.0527 | 0.6931 | 0.2737 | 7.646 |
| 8 | 1 | 0.0475 | -1.0525 | 0.6914 | 0.2717 | 7.589 |
| 20 | 1 | 0.0478 | -1.0522 | 0.6909 | 0.2713 | 7.578 |

It is seen from these tables that $r = 0$, corresponding to ordinary Padé approximation, does not yield very good models in terms of $SES_{rel}$ values. However, for the third order models with $r \geq 2$ excellent models are produced by the LS method, and similarly for the second order models with $r \geq 4$.

For comparison, the true optimal third and second order models (Lucas 1993d) in terms of minimum SES values are given by

$$G_3(z) = \frac{0.9965z^2 - 0.4692z - 0.3967}{z^3 - 1.5909z^2 + 1.1100z - 0.3034}$$

with SES = 0.0017 and $SES_{rel}$ = 0.048% and

$$G_2(z) = \frac{1.1754z - 0.3331}{z^2 - 1.1441z + 0.6871}$$

with SES = 0.2077 and $SES_{rel}$ = 5.802%. It is seen that the LS Padé models are not much inferior to the true optimal ones and are achieved by much less calculation.

**Step Response Models**

It should further be noticed that the technique can be readily applied for system inputs other than the unit pulse by using the transfer function of the *transient part* of the response (Lucas 1993d). For example, if the input is a unit step function then the transfer function becomes

$$T(z) = \frac{\{G(z) - G(1)\}z}{z - 1} = zX(z)$$

and $X(z)$ is reduced to $X_k(z)$, giving the final reduced model as

$$G_k(z) = (z-1)X_k(z) + G(1)$$

which ensures the steady-state matching of responses.

**Example 6.3**

This approach is demonstrated using once more the fourth order $z$-transfer function

$$G(z) = \frac{z^3 - 0.1z^2 - 0.47z - 0.225}{z^4 - 1.2z^3 + 0.55z^2 + 0.05z - 0.075}$$

Tables 6.3 and 6.4 give information about the third and second order reduced models produced in this way, including the square error sum (SES) and *relative* square error sum ($SES_{rel}$) values, defined by

$$SES = \sum_{i=0}^{\infty}(y_i - \bar{y}_i)^2 \qquad \text{and} \qquad SES_{rel} = \frac{\sum_{i=0}^{\infty}(y_i - \bar{y}_i)^2}{\sum_{i=0}^{\infty}(y_i - G(1))^2}$$

where the $y_i$ and $\bar{y}_i$ are the respective full and reduced models' $i$th step response values.

**Table 6.3**

| $r$ | $d_2$ | $d_1$ | $d_0$ | $e_2$ | $e_1$ | $e_0$ | SES | $SES_{rel}$% |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | -0.7199 | -0.2185 | -1.8199 | 1.4834 | -0.5658 | 1.594 | 14.878 |
| 2 | 1 | -0.5472 | -0.3255 | -1.6472 | 1.1996 | -0.3506 | 0.017 | 0.220 |
| 4 | 1 | -0.5273 | -0.3409 | -1.6273 | 1.1666 | -0.3304 | 0.008 | 0.100 |
| 6 | 1 | -0.5269 | -0.3414 | -1.6269 | 1.1659 | -0.3302 | 0.008 | 0.100 |
| 8 | 1 | -0.5264 | -0.3418 | -1.6264 | 1.1650 | -0.3297 | 0.0076 | 0.098 |
| 20 | 1 | -0.5263 | -0.3419 | -1.1663 | 1.1650 | -0.3296 | 0.0076 | 0.098 |

**Table 6.4**

| $r$ | $d_1$ | $d_0$ | $e_1$ | $e_0$ | SES | $SES_{rel}$% |
|---|---|---|---|---|---|---|
| 0 | 1 | -0.4601 | -1.5601 | 1.4161 | unstable | - |
| 2 | 1 | -0.7480 | -1.5418 | 0.9413 | 13.541 | 173.8 |
| 4 | 1 | -0.7800 | -1.4465 | 0.7952 | 1.641 | 21.1 |
| 6 | 1 | -0.7770 | -1.4370 | 0.7905 | 1.592 | 20.4 |
| 8 | 1 | -0.7775 | -1.4347 | 0.7875 | 1.560 | 20.0 |
| 20 | 1 | -0.7774 | -1.4341 | 0.7871 | 1.557 | 20.0 |

It is seen from these tables that the same pattern of improvement on the results of ordinary Padé approximation is observed in the step response case as is observed in that of the pulse response illustrated earlier and for the same low values of $r$.

Again, for comparison the true optimal third and second order step response models (Lucas 1993d) in terms of minimum SES values are given by

$$G_3(z) = \frac{0.0021z^3 + 0.984z^2 - 0.3974z - 0.4419}{z^3 - 1.5493z^2 + 1.0557z - 0.2737}$$

with SES = 0.000915 and $SES_{rel} = 0.017\%$ and

$$G_2(z) = \frac{-0.2379z^2 + 1.8984z - 1.4745}{z^2 - 1.3057z + 0.6005}$$

with SES = 0.486 and $SES_{rel} = 6.237\%$. It is seen that the accuracy of the second order LS Padé model is too high to be useful, but is close enough to the accuracy of the corresponding true optimal model to again indicate that satisfactory models may be achieved by this method with much less computation than the optimal method.

## 6.3 A Stability Preserving LS Method for Continuous-time Systems

It was seen in section 3.2 that the possibility of producing unstable reduced models from stable systems is a serious drawback of classical Padé approximation. In the present section it is shown how the stability preservation property proven in section 6.2 for discrete-time systems may be used to guarantee the stability of reduced models of continuous-time systems also.

It is well-known (Chen *et al*, 1979) that simple bilinear transforms may be used to transfer discrete-time systems from the $z$-domain to the $s$-domain and vice-versa. In particular, the bilinear transform

$$sT = \frac{z-1}{z+1}$$

($T$ is the sampling period) maps the points of the left half-plane in the $s$-domain on to points inside the unit disc in the $z$-domain. Hence, any transfer function representing a stable continuous-time system will have its poles mapped into the unit circle in the $z$-domain and so will also be stable as a discrete-time system.

Thus

$$G(s) = \frac{b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \cdots + b_0}{a_n s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \cdots + a_0}$$

will map into the $z$-transfer function

$$H(z) = \frac{\hat{b}_n z^n + \hat{b}_{n-1}z^{n-1} + \cdots + \hat{b}_1 z + \hat{b}_0}{\hat{a}_n z^n + \hat{a}_{n-1}z^{n-1} + \hat{a}_{n-2}z^{n-2} + \cdots + \hat{a}_0}$$

In the same way, the inverse bilinear transform

$$z = \frac{1+sT}{1-sT}$$

can be used to produce transfer functions of stable continuous-time systems from the $z$-transfer functions of stable discrete-time systems. In what follows the value of $T$ is taken to be unity without loss of generality.

All the elements of a new stability preserving model reduction technique for continuous-time are now in place. First there is the bilinear transformation of the high-order transfer function of the full system to a stable high-order $z$-transfer function. This is followed by the reduction of this $n$th order $z$-transfer function to $k$th order by LS Padé approximation using Markov parameters only. The property proved in the last section guarantees ($e_k = 1$) that the reduced $z$-transfer function will be stable. This guarantees in turn that the inverse bilinear transformation on the

denominator polynomial of this $z$-transfer function will produce a reduced stable denominator polynomial in the continuous-time transfer function. The numerator polynomial may then be found by moment and/or Markov parameter matching. The inverse bilinear transformation is applied only to the denominator of the reduced $z$-transfer function as the accuracy of the final reduced continuous-time system is likely to benefit from the inclusion of $k$ items of information from the full system.

**Example 6.2**

So that a ready comparison may be made with the stability preservation techniques considered in chapter 3, the sixth-order system given by the transfer function

$$G(s) = \frac{s^5 + 17.5s^4 + 111s^3 + 314.5s^2 + 388s + 168}{s^6 + 15s^5 + 93s^4 + 307s^3 + 562s^2 + 562s + 260}$$

will be approximated by third and second-order models using the above method.

From past experience, good stable reduced order models can often be generated by the LS method for discrete-time systems when $r = 2$ or 3 only. However, a good "rule-of-thumb" criterion to use is

$$r = 2(n - k) \qquad (6.15)$$

where $n$ and $k$ are the orders of the full and reduced systems respectively. It is noted that all the information for the full $n$th order system is contained in the first $2n + 1$ Markov parameters and, therefore, it is reasonable to expect that the reduced order model will often require *at most* this amount of information to enable an adequate approximation. Equating $2k + r + 1$ to $2n + 1$ then gives the criterion in (6.15) which has been found to be quite robust. This criterion has been applied in the following example to select the value of $r$ for each model reduction performed.

The bilinear transform

$$s = \frac{z-1}{z+1} \qquad (T=1)$$

gives the $z$-transfer function

$$H(z) = \frac{1000z^6 + 3150z^5 + 3800z^4 + 2172z^3 + 576z^2 + 54z}{1800z^6 + 4680z^5 + 5224z^4 + 3304z^3 + 1296z^2 + 304z + 32}$$

and the LS Padé method using only Markov parameters applied to $H(z)$ provides the third order denominator polynomial ($r = 6$)

$$z^3 + 1.29887z^2 + 0.60608z + 0.17007$$

The third order model derived by inverse bilinear transform

$$z = \frac{1+s}{1-s}$$

and matching the first three time moments for the numerator is

$$R_3(s) = \frac{0.1307411s^2 + 2.3504739s + 1.9869362}{0.137139s^3 + 1.60525s^2 + 3.18258s + 3.07502}$$

with

$$I_{rel} = 0.33\% \qquad \text{and} \qquad J_{rel} = 0.13\%$$

{Notice that the continuous-time reduced order denominator is obtained by mapping the poles of the reduced system in the $z$-domain back to the $s$-domain, which in this case is the expression

$$(1-s)^3 + 1.29887(1-s)^2(1+s) + 0.60608(1-s)(1+s)^2 + 0.17007(1+s)^3\}.$$

The accuracy of this approximation is an improvement on that achieved by all three of the more traditional stability-preserving methods of section 3.3. Also, the same technique produces the second order transfer function ($r = 8$)

$$R_2(s) = \frac{1.1379849s + 1.1291216}{0.749959s^2 + 1.50257s + 1.74745}$$

with

$$I_{rel} = 2.39\% \qquad \text{and} \qquad J_{rel} = 0.61\%$$

which matches or improves upon the accuracy of the second order models found using the same three stability preservation methods.

When this LS stability preservation method is applied to the same model with a value 0.5 for the sampling period $T$, the third order model produced is

$$R_3(s) = \frac{0.12245s^2 + 1.07198s + 0.8288345}{0.0954775s^3 + 0.75376s^2 + 1.4692s + 1.28272}$$

with

$$I_{rel} = 1.11\% \qquad \text{and} \qquad J_{rel} = 0.52\%$$

and the second order model is

$$R_2(s) = \frac{0.57713s + 0.6580302}{0.374162s^2 + 0.742479s + 1.01838}$$

with

$$I_{rel} = 2.77\% \qquad \text{and} \qquad J_{rel} = 2.24\%$$

indicating that sometimes no improvement is to be gained for the method by expending effort in varying the value of $T$.

It is seen that this new bilinear transform LS stability preservation method can give results as good as existing methods and with less computational effort than either Routh approximation or the Stability Equation method.

It is interesting that Lucas and Beat (1990) proposed a linear shift $s \rightarrow s + a$ be used to overcome problems of unstable reduced models produced by LS Padé moment matching. However, despite this and the fact that LS Padé moment matching

with linear shift can sometimes produce reduced models of excellent accuracy when stability is achieved, it does *not* provide the guaranteed stability offered by the bilinear transform LS method.

The graphs in figures 6.1 and 6.2 allow a visual comparison of the impulse and step responses of the reduced models and the full system for a sampling period of one unit. The equivalent results are shown in figures 6.3 and 6.4 for the models obtained by this method using a sampling period of $T = ½$.

## 6.4 Generalised LS Padé Method for Discrete-time Systems

In this section the generalised LS method for the reduction of continuous-time systems is extended to the reduction of discrete-time systems. However, it is shown here that the two-stage analysis cannot be implemented in the discrete-time case due to a complication affecting the use of both time moment and Markov parameter information in the same calculation.

A reduced $k$th order model is to be found by the LS Padé method for

$$G(z) = \frac{b_n z^n + b_{n-1} z^{n-1} + \cdots + b_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_0}$$

using the system information from the first $2k + r$ Markov parameters and the first $t$ time moments where for simplicity it is assumed that $r > 0$ and $t < k$. Suppose that $G(z)$ is reduced to the system given by

$$R(z) = \frac{d_k z^k + d_{k-1} z^{k-1} + \cdots + d_1 z + d_0}{z^k + e_{k-1} z^{k-1} + \cdots + e_1 z + e_0}$$

The Markov parameters are obtained by expanding $G(z)$ about $z = \infty$, and equating $R(z)$ gives the $2k + r + 1$ equations

**Fig 6.1 Sample Period T=1**

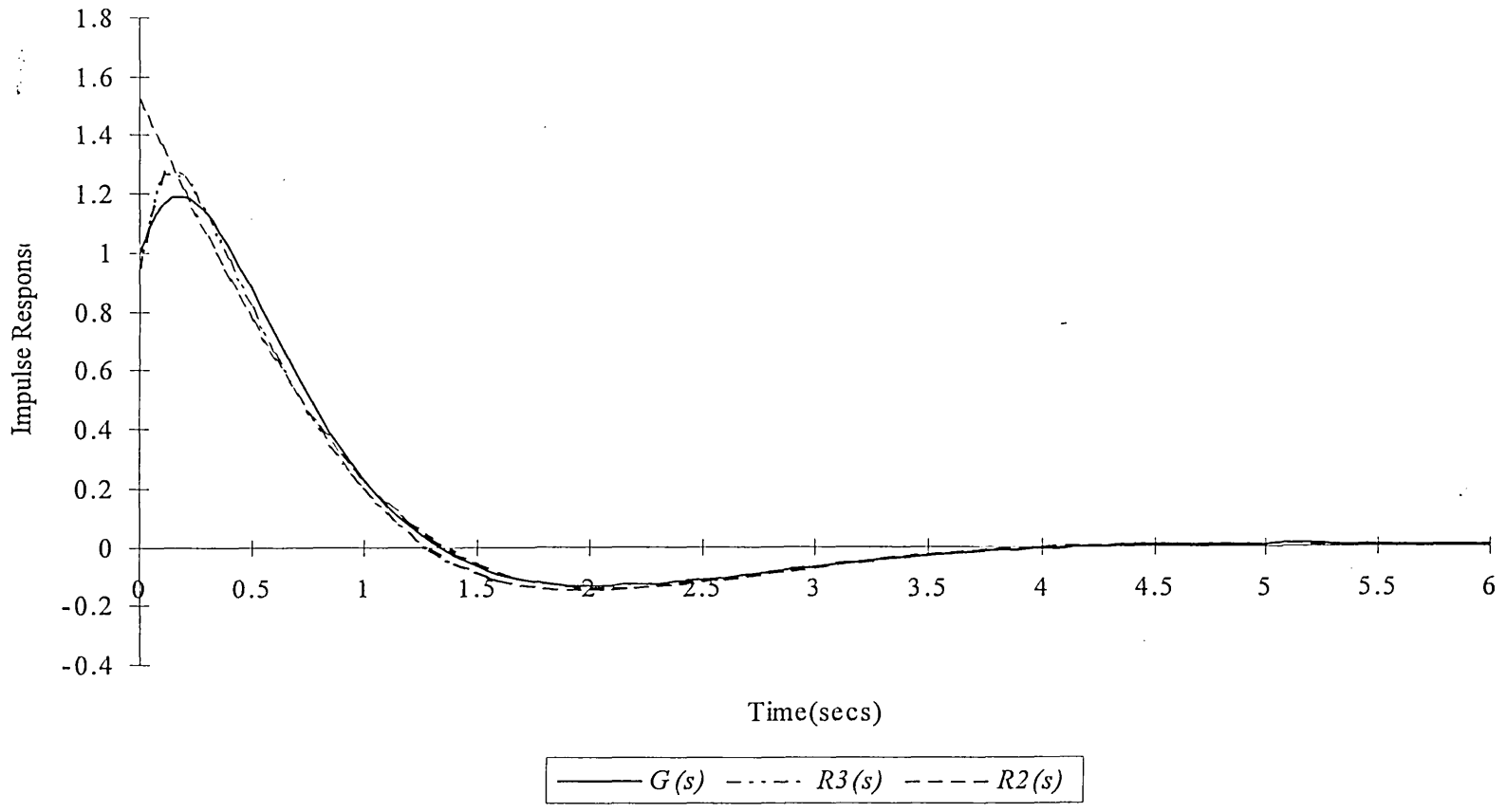


Fig 6.2 Sample Period T=1

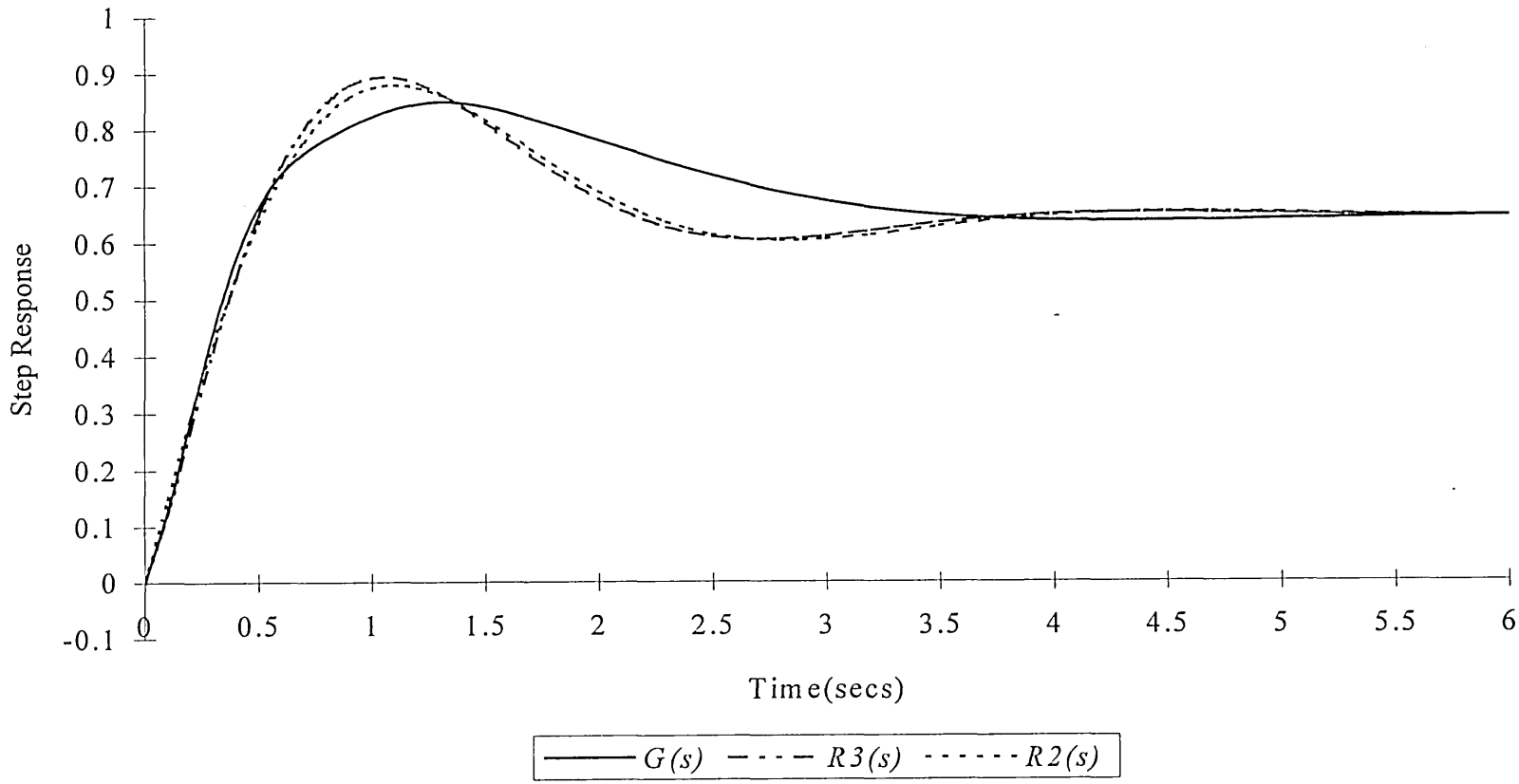


**Fig 6.3 Sample Period T=0.5**

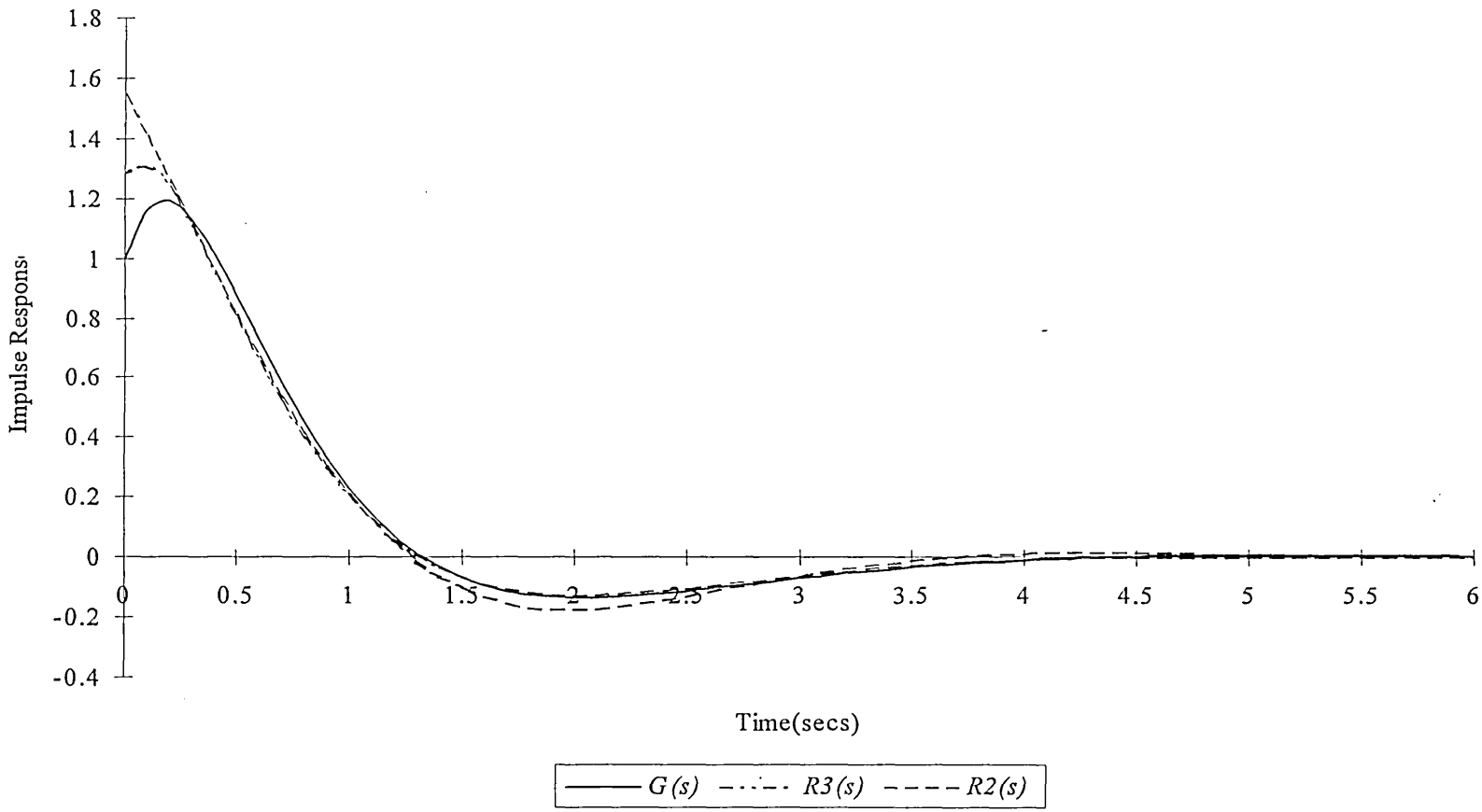
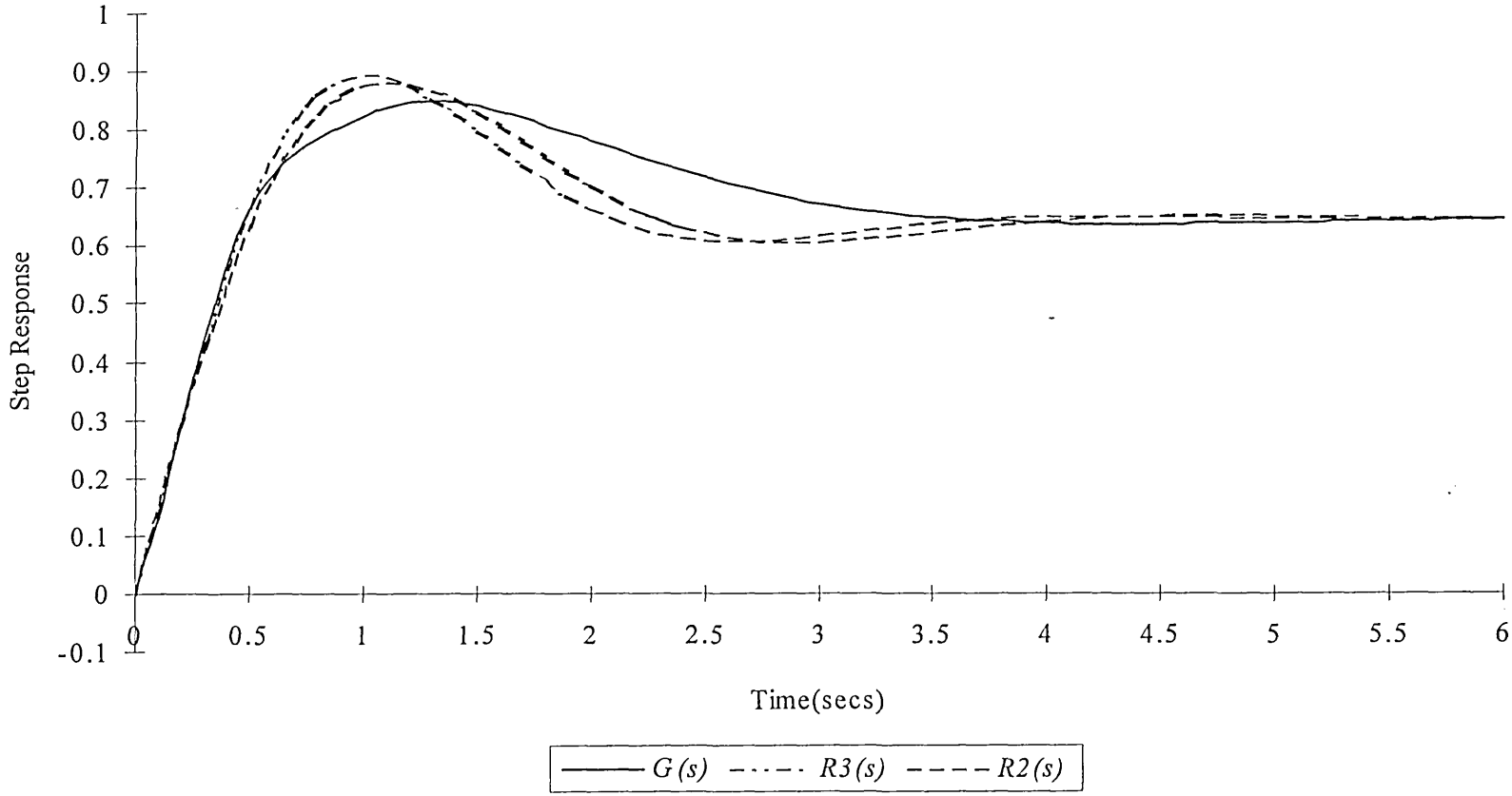


**Fig 6.4 Sample Period T=0.5**



$$
\begin{aligned}
d_k &= m_0 \\
d_{k-1} &= m_0 e_{k-1} + m_1 \\
&\vdots \\
d_0 &= m_0 e_0 + m_1 e_1 + \cdots + m_{k-1} e_{k-1} + m_k \\
0 &= m_1 e_0 + m_2 e_1 + \cdots + m_k e_{k-1} + m_{k+1} \\
0 &= m_2 e_0 + m_3 e_1 + \cdots + m_{k+1} e_{k-1} + m_{k+2} \\
&\vdots \\
0 &= m_{k+r} e_0 + m_{k+r+1} e_1 + \cdots + m_{2k+r-1} e_{k-1} + m_{2k+r}
\end{aligned}
\qquad (6.16)
$$

This parallels the LS method of the continuous-time case. It is when considering the $t$ equations to be utilised for the time moment information that a problem for the two-stage analysis becomes apparent.

The time moments $c_i$ ($i = 0, 1, 2, \ldots$) of a discrete-time system are defined (Shamash and Feinmesser 1978) by the expansion of $G(z)$ about $z = 1$, i.e.

$$G(z) = c_0 + c_1(z-1) + c_2(z-1)^2 + \ldots$$

Therefore, the Taylor series expansion method for the calculation of the first $k$ time moments of the discrete-time system requires the application of a linear shift $z = p + 1$. For this reason, to form a matrix-vector equation combining both time moment and Markov parameter information in the discrete-time case, transformation matrices would be required to express the Markov parameters of $H(p) = G(p + 1)$ in terms of those of $G(z)$. The nature of the transformation matrices required can be seen from the relationship between the Markov parameters of $H(p)$ and $G(z)$.

Taking the Markov parameters of $G(z)$ to be $m_i$ and those of $H(p)$ to be $\alpha_i$ ($i = 1, 2, \ldots$) and as

$$G(z) = G(p+1)$$

we have the relationship

$$\frac{m_1}{z}+\frac{m_2}{z^2}+\frac{m_3}{z^3}+\cdots=\frac{\alpha_1}{(z-1)}+\frac{\alpha_2}{(z-1)^2}+\frac{\alpha_3}{(z-1)^3}+\cdots \qquad (6.17)$$

Expressing the general term of the series on the R.H.S. of (6.17) in the form

$$\frac{\alpha_i}{z^i}\left(1-\frac{1}{z}\right)^{-i} \qquad i=1,2,\ldots$$

and expanding each term using the Binomial Theorem before collecting terms in $z^{-i}$ gives the set of equations

$$\begin{aligned}
\alpha_1 &= m_1 \\
\alpha_2 &= -m_1+m_2 \\
\alpha_3 &= m_1-2m_2+m_3 \\
\alpha_4 &= -m_1+3m_2-3m_3+m_4 \\
&\vdots
\end{aligned}$$

relating the two sets of Markov parameters. It is clear from these equations that the transformation matrices required for the expression of the $\alpha_i$ in terms of the $m_i$ will have lower triangular form.

It was noted in section 5.6 that, for an equivalent partial LS method to be established, the matrix to be inverted for the two-stage analysis must be of the form

$$\left[\begin{array}{c|c} f_1 I_r & \Phi \\ \hline \Phi & f_2 I_{k-r} \end{array}\right]$$

where $f_1$ and $f_2$ are scalar factors. However, premultiplication of the matrices containing the $H(p)$ Markov parameters by lower triangular transformation matrices does not produce a matrix for inversion in the two-stage analysis of this required form. Consequently, no equivalent partial LS method can be established in this case.

## 6.5 Mixed Moment and Markov Parameter Matching

The complication involved in combining time moment and Markov parameter information in the same matrix-vector equation can be avoided, however, if attention is restricted to a special case of the generalised LS-Padé method for discrete-time systems. This may be achieved by employing a partial approach where Markov parameters only are used to determine the denominator coefficients of the reduced order model followed by exact moment matching for the numerator coefficients.

Therefore, the first stage in the calculation would consist of solving the last $k + r$ equations of (6.16) in the matrix-vector form (6.4)

$$H^T H\mathbf{e} = H^T\mathbf{q}$$

to find the reduced denominator. This would be followed by matching the expansions of $R(z)$ and $G(z)$ about the point $z = 1$ for the first $k + 1$ terms to give the reduced numerator. Neither of these calculations necessitates the use of transformation matrices to express system parameter information in the correct form. Also, it allows us to make use of the stability preservation property proved for Markov parameter matching in section (6.2) by choosing to set $e_k = 1$.

# CHAPTER 7

# EXTENSION OF LEAST-SQUARES PADÉ METHODS TO MULTIVARIABLE SYSTEMS

## 7.1 Introduction

For the extension of the ideas presented for single input/ single output (SISO) systems in chapters 4 – 6 to multivariable systems it is necessary to apply the Laplace transform as defined in chapter 1 to the state equations

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$$
$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t)$$

representing the general linear, time-invariant system. The equations become (Sinha 1984)

$$s\mathbf{X}(s) - x(0) = A\mathbf{X}(s) + B\mathbf{U}(s)$$

$$\mathbf{Y}(s) = C\mathbf{X}(s) + D\mathbf{U}(s)$$

which may be written in the form

$$\mathbf{Y}(s) = \mathbf{G}(s)\mathbf{U}(s)$$

where $\mathbf{x}(0)$ is assumed to be zero without loss of generality. For a multivariable system with $m$ inputs and $l$ outputs $\mathbf{G}(s)$ is the $l$ x $m$ matrix given by

$$\mathbf{G}(s) = C(sI - A)^{-1}B + D$$

and is known as the *transfer function matrix* of the system $T(A, B, C, D)$. Therefore, in the frequency domain the representation of the system is in terms of a matrix of rational functions

$$\mathbf{G}(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) & \cdots & G_{1m}(s) \\ G_{21}(s) & G_{22}(s) & \cdots & G_{2m}(s) \\ \vdots & \vdots & & \vdots \\ G_{l1}(s) & G_{l2}(s) & \cdots & G_{lm}(s) \end{bmatrix}$$

where each of the elements $G_{ij}(s)$ represent the transfer function of the SISO subsystem connecting the $j$th input and the $i$th output of the multivariable system.

Some work on the application of exact Padé methods (Shamash 1976) and LS Padé methods (Aguirre 1995) has been done already in the literature. However, the ease with which the general multivariable system can be represented in terms of its various SISO subsystems should not blind us to certain complications affecting the extension of LS Padé methods of model reduction to the multivariable case. Some theoretical considerations concerning transfer function matrices are discussed in the following section. These will clarify what is meant by order reduction in relation to multivariable systems and help in the clear exposition of procedures for the application LS Padé approximation in this case.

## 7.2 Order Reduction of Multivariable Systems

In the case of the general $n$th order SISO system

$$G(s) = \frac{b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \cdots + b_0}{a_n s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \cdots + a_0}$$

the definition of order reduction is straightforward and intuitive. The order of a transfer function has been reduced if it is approximated by a proper rational function

$$R(s) = \frac{d_{k-1}s^{k-1} + \cdots + d_1 s + d_0}{e_k s^k + e_{k-1}s^{k-1} + \cdots + e_1 s + e_0}$$

where $k$ the order of the denominator polynomial is less than $n$ and the corresponding linear, time-invariant system is realisable. Such a simple account of order reduction is not available to us in the multivariable case because of the possible different denominator polynomials of the elements of the transfer function matrix.

**The Poles and Zeros of the Transfer Function Matrix**

Consider the clear definition of what is meant by a system zero in the multivariable case. It is noted that, whereas in the SISO case the transfer function $G(s)$ has a zero at $s = a$ if $G(a) = 0$, it is *not* the case that a zero of the transfer function matrix $\mathbf{G}(s)$ is a value which makes it a null matrix. In fact (Sinha 1984), in the multivariable case, a zero of $\mathbf{G}(s)$ is a value which reduces the rank of the transfer function matrix below its normal rank, i.e. for which local rank is less than normal rank.

In the SISO case, $G(s)$ is said to have a pole at $s = a$ if $G(a) = \infty$. Taking the multivariable case, it is noticed (Sinha 1984) that if the system's eigenvalues are distinct, denoted by $\lambda_i$, the transfer function matrix may be written as

$$\mathbf{G}(s) = \sum_{i=1}^{n} \frac{\mathbf{G}_i}{(s - \lambda_i)} \qquad \text{where} \qquad \mathbf{G}_i = \lim_{s \to \lambda_i} (s - \lambda_i)\mathbf{G}(s)$$

and $\mathbf{G}_i$ is known as the *residue matrix* of $\mathbf{G}(s)$ at $\lambda_i$. This means that

$$\lim_{s \to \lambda_i} \|\mathbf{G}(s)\| = \lim_{s \to \lambda_i} \left\| \sum_{i=1}^{n} \frac{\mathbf{G}_i}{s - \lambda_i} \right\| = \infty$$

which provides the basis of a natural extension of the notion of a system pole in the SISO case to that of a transfer function matrix. A number $a$ which may be complex is said to

be a *pole* of order $n$ of $\mathbf{G}(s)$ if some element of $\mathbf{G}(s)$ has a pole order $n$ at $s = a$ ( in the SISO sense) and no element has a pole of order larger than $n$ at $s = a$.

Care must be exercised in the interpretation of this definition of system pole in the $\mathbf{G}(s)$ of some multivariable case. Since there may be cancellations between the numerator and denominator polynomials in the expressions for the proper rational function elements of $\mathbf{G}(s)$, the set of poles given by

$$\mathbf{G}(s) = \frac{C\text{adj}(sI - A)B}{\det(sI - A)}$$

is generally a subset of the set of eigenvalues obtained from solving the characteristic equation

$$\det(sI - A) = 0$$

This distinction is brought out by referring to the eigenvalues as the *characteristic frequencies*, being the roots of the *characteristic polynomial* of the system matrix. In contrast the poles are said to be the roots of the *pole polynomial* which is the lowest common denominator (L.C.D.) of all the non-identically zero minors of all orders of the transfer function matrix (MacFarlane 1974).

This indicates most importantly that the order of a multivariable system is not simply defined as the degree of the L.C.D. polynomial of $\mathbf{G}(s)$ since such polynomials may vary in their degree depending on the form in which the transfer function matrix is expressed. Therefore, $\mathbf{G}(s)$ must be expressed in an appropriate form if the order of the system is to be identified with certainty depending on the form and an example is given below to demonstrate this point.

**Smith-McMillan Canonical Form**

An appropriate form (MacFarlane 1974, Sinha 1984) in which to express the transfer function matrix if we wish to identify the order of a system with certainty is the Smith-McMillan form for matrices of rational functions. It has been developed from the Smith form for polynomial matrices (Rosenbrock 1970).

$\mathbf{G}(s)$ has been defined as an $l$ x $m$ proper transfer function matrix representing a multivariable system having $m$ inputs and $l$ outputs. It may be expressed in the form

$$\mathbf{G}(s) = L(s)M(s)R(s) \qquad (7.1)$$

where $L(s)$ and $R(s)$ are $l$ x $l$ and $m$ x $m$ respectively and are both unimodular (having determinants which are independent of $s$). In Smith-McMillan form

$$M(s) = \left[\begin{array}{c|c} \mathrm{diag}\left\{\dfrac{\psi_i(s)}{\phi_i(s)}\right\} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right]$$

where the top, left-hand partition is a diagonal $r$ x $r$ matrix containing transfer function elements such that $\psi_i(s)$ is a factor of $\psi_{i+1}(s)$, $\phi_i(s)$ is a factor of $\phi_{i-1}(s)$ and $r$ is the normal rank of $\mathbf{G}(s)$. Once in this form the *McMillan degree* of the transfer function matrix may be calculated by the sum of the degrees of all the $\phi_i(s)$ of $M(s)$. The McMillan degree is the same as that of the pole polynomial defined as the L.C.D. of all the non-identically zero minors of all orders. This can be stated with certainty to be the *order* of the multivariable system associated with the transfer function matrix. The following example (MacFarlane 1974, Sinha 1984) illustrates this point.

**Example 7.1**

Consider the transfer function matrix

$$G(s)=\begin{bmatrix} \dfrac{1}{s+1} & 0 & \dfrac{s-1}{(s+1)(s+2)} \\ -\dfrac{1}{s-1} & \dfrac{1}{s+2} & \dfrac{1}{s+2} \end{bmatrix}$$

when expressed in Smith-McMillan form has

$$M(s)=\begin{bmatrix} \dfrac{1}{(s+1)(s+2)(s-1)} & 0 & 0 \\ 0 & \dfrac{s-1}{s+2} & 0 \end{bmatrix}$$

where $L(s)$ and $R(s)$ are the unimodular matrices

$$L(s)=\begin{bmatrix} (s-1) & \frac{1}{2} \\ -\frac{1}{3}(2s^2+3s+1) & -\frac{1}{6}(2s+5) \end{bmatrix}$$

$$R(s)=\begin{bmatrix} \frac{1}{3}(s+2)(4-s^2) & -\frac{1}{2}(s-1)(s+1) & -\frac{1}{3}(s-1)(s^2+3s-1) \\ \frac{2}{3}(s+2) & 1 & \frac{2}{3}(s+2) \\ 1 & 0 & 1 \end{bmatrix}$$

From inspection of the denominators of the diagonal elements of $M(s)$ it is seen that the McMillan degree is 4 which is also the degree of the pole polynomial of $\mathbf{G}(s)$ identified by consideration of the minors of all orders

$$p(s)=(s+1)(s+2)^2(s-1)$$

On the other hand, the L.C.D. of both the Smith-McMillan form and the original expression of $\mathbf{G}(s)$ is given by $(s+1)(s+2)(s-1)$ which is of degree 3. This illustrates

clearly that the L.C.D. of the elements of $\mathbf{G}(s)$ does not necessarily give an accurate indication of the order of the multivariable system concerned. Hence, in any examples where model reduction of a multivariable system is attempted, the McMillan degree of the simplified model must be calculated before it can be ascertained whether *any actual reduction of the order of the full system has been achieved.*

In practice, MacFarlane points out that finding the McMillan degree via the Smith-McMillan form is extremely awkward when carried out manually. The natures of the matrices $L(s)$ and $R(s)$ in example 7.1 give an indication of the kind of problems even in a relatively simple case. A more practical approach is to use the fact that the McMillan degree is the same as that of the pole polynomial $p(s)$ defined as the L.C.D. of all the non-identically zero minors of all orders. This is the approach that will be used for any examples in this chapter to determine whether true order reduction has been achieved.

## 7.3 Matrix Fraction Descriptions

To facilitate the clear description of procedures for the application of LS Padé methods to multivariable systems a close analogy to SISO systems is developed (Sinha 1984) by recognising that a transfer function matrix may be factorised into the product of a polynomial matrix and the inverse of another polynomial matrix. Because of the lack of commutativity this leads to the definition of two possible matrix fraction descriptions of the general transfer function matrix $\mathbf{G}(s)$:

the right-hand matrix description (R.M.D.)

$$\mathbf{G}(s) = N_1(s)D_1^{-1}(s) \tag{7.2}$$

and the left-hand matrix description (L.M.D.)

$$\mathbf{G}(s) = D_2^{-1}(s)N_2(s) \tag{7.3}$$

It is noted that the matrices involved in these descriptions can be expressed in terms of the Smith-McMillan form dealt with in section 7.2 so that

$$N_1(s)D_1^{-1}(s) = D_2^{-1}(s)N_2(s) = L(s)M(s)R(s)$$

where

$$M(s) = \left[\begin{array}{c|c} \mathrm{diag}\left\{\dfrac{\psi_i(s)}{\phi_i(s)}\right\} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right]$$

Further, it is seen that because of the diagonal nature of the non-zero partition of the matrix it may be written as either

$$M(s) = M_\psi(s)\left[M_{\phi R}(s)\right]^{-1} \tag{7.4}$$

or

$$M(s) = \left[M_{\phi L}(s)\right]^{-1}M_\psi(s) \tag{7.5}$$

where

$$M_\psi = \left[\begin{array}{c|c} \mathrm{diag}\{\psi_i(s)\} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right]$$

$$M_{\phi R}(s) = \left[\begin{array}{c|c} \mathrm{diag}\{\phi_i(s)\} & \mathbf{0} \\ \hline \mathbf{0} & I \end{array}\right] \qquad M_{\phi L} = \left[\begin{array}{c|c} \mathrm{diag}\{\phi_i(s)\} & \mathbf{0} \\ \hline \mathbf{0} & I \end{array}\right]$$

all the diagonal blocks are of dimension $r$ x $r$ (governed by the normal rank of $G(s)$), $M_\psi(s)$ is $l$ x $m$, $M_{\phi R}(s)$ is $m$ x $m$ and $M_{\phi L}(s)$ is $l$ x $l$.

For the R.M.D. of $\mathbf{G}(s)$ we have

$$N_1(s)D_1^{-1}(s) = L(s)M(s)R(s)$$

and substituting the expression for $M(s)$ given in (7.4) gives

$$N_1(s)D_1^{-1}(s) = L(s)M_\psi(s)\left[M_{\phi R}(s)\right]^{-1}R(s) = L(s)M_\psi(s)\left[R^{-1}(s)M_{\phi R}(s)\right]^{-1}$$

Hence for the R.M.D., since there is a unique Smith-McMillan form of $\mathbf{G}(s)$, we have

$$N_1(s) = L(s)M_\psi(s) \qquad \text{and} \qquad D_1(s) = R^{-1}(s)M_{\phi R}(s)$$

In a similar way, using (7.5), the expressions for the numerator and denominator polynomial matrices in the L.M.D. can be written as

$$N_2(s) = M_\psi(s)R(s) \qquad \text{and} \qquad D_2(s) = M_{\phi L}(s)L^{-1}(s)$$

Therefore, both of the matrix fraction descriptions of $\mathbf{G}(s)$ can be seen to be uniquely expressed in terms of the Smith-McMillan form. Further it is noted that for the R.M.D.

$$\det\{D_1(s)\} = \det\{R^{-1}(s)\}\det\{M_{\phi R}(s)\} = \alpha\prod_{i=1}^{r}\phi_i(s) = \alpha p(s)$$

and for the L.M.D.

$$\det\{D_2(s)\} = \det\{M_{\phi L}(s)\}\det\{L^{-1}(s)\} = \beta\prod_{i=1}^{r}\phi_i(s) = \beta p(s)$$

where the unimodular nature of the matrices $R^{-1}(s)$ and $L^{-1}(s)$ means that $\alpha$ and $\beta$ are scalars independent of $s$; $p(s)$ is identically equal to the pole polynomial defined in

section 7.2. This leads to the observation that the poles of the multivariable system are the roots of either of the equations

$$\det\{D_1(s)\} = 0 \qquad \text{or} \qquad \det\{D_2(s)\} = 0$$

## 7.4 Multivariable Least-squares Padé Approximation

In this section the idea of matrix fraction descriptions is used to give a clear exposition of the application of LS Padé approximation in the multivariable case. As for the SISO case, the LS Padé approach is an extension of the exact Padé method, the latter being first applied to multivariable systems by Shamash (1976). It is seen that the application of these methods to multivariable systems gives rise to a number of issues that are addressed in this section. Finally, examples are included to illustrate a number of important points about the multivariable application of this method.

### The Exact Padé Method

The exact Padé method was described in Shamash (1976). This description is now expanded in the light of more recent work (Bandyopadhyay and Lamba 1987). Further, the idea of matrix fraction descriptions is used to highlight important factors affecting the application of this method and its LS extension to the multivariable case.

It is noted that the expansion of the full system $\mathbf{G}(s)$ in terms of the time moment matrices $C_i$ ($i = 0, 1, 2, \ldots$) is given by

$$\mathbf{G}(s) = C_0 + C_1 s + C_2 s^2 + \ldots$$

A simplified system $\mathbf{R}(s)$ may be derived by matching an appropriate number of terms of this expansion for the full and simplified systems. However, it is seen in section 7.3 that

there are two approaches that may be taken to the description of $\mathbf{R}(s)$ corresponding to the R.M.D. and L.M.D. described there.

Let us consider first the R.M.D. approach using

$$\mathbf{R}(s) = \hat{N}_1(s)\hat{D}_1^{-1}(s)$$

where

$$\hat{N}_1(s) = A_0 + A_1 s + A_2 s^2 + \ldots + A_{d-1} s^{d-1}$$

$$\hat{D}_1(s) = B_0 + B_1 s + B_2 s^2 + \ldots + B_{d-1} s^{d-1} + I s^d$$

Bandyopadhyay and Lamba (1987) show that the appropriate number of terms to match for a system with $l$ outputs and $m$ inputs when finding a $k$th order Padé approximant is given by

$$p = \frac{k}{l} + \frac{k}{m}$$

where both $k/l$ and $k/m$ are integer. Therefore the R.M.D. approach method uses the first $p$ equations of the infinite set given by

$$\hat{N}_1(s) = \left(C_0 + C_1 s + C_2 s^2 + \ldots\right)\hat{D}_1$$

which may be written as

$$\begin{aligned}
A_0 &= C_0 B_0 \\
A_1 &= C_1 B_0 + C_0 B_1 \\
&\vdots \\
A_{\frac{k}{m}-1} &= C_{\frac{k}{m}-1} B_0 + C_{\frac{k}{m}-2} B_1 + \ldots + C_0 B_{\frac{k}{m}-1} \\
\mathbf{0} &= C_{\frac{k}{m}} B_0 + C_{\frac{k}{m}-1} B_1 + \ldots + C_1 B_{\frac{k}{m}-1} + C_0 \\
&\vdots \\
\mathbf{0} &= C_{p-1} B_0 + C_{p-2} B_1 + \ldots + C_{\frac{k}{l}} B_{\frac{k}{m}-1} + C_{\frac{k}{l}-1}
\end{aligned} \qquad (7.6)$$

This may be written in the matrix form

$$HX = C \tag{7.7}$$

where

$$H = \begin{bmatrix} I_m & \mathbf{0} & \cdots & \mathbf{0} & -C_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & I_m & \cdots & \mathbf{0} & -C_1 & -C_0 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & I_m & -C_{\frac{k}{m}-1} & -C_{\frac{k}{m}-2} & \cdots & -C_0 \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -C_{\frac{k}{m}} & -C_{\frac{k}{m}-1} & \cdots & -C_1 \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -C_{\frac{k}{m}+1} & -C_{\frac{k}{m}} & \cdots & -C_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -C_{p-1} & -C_{p-2} & \cdots & -C_{\frac{k}{l}} \end{bmatrix}$$

$$X = \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_{d-1} \\ B_0 \\ B_1 \\ \vdots \\ B_{d-1} \end{bmatrix} \qquad \text{and} \qquad C = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ C_0 \\ C_1 \\ \vdots \\ C_{\frac{k}{l}-1} \end{bmatrix}$$

The relationship between the degree, $d$, of the polynomial elements of the denominator matrix and the order, $k$, of the Padé approximant is given by

$$d = \frac{k}{l} \qquad \text{for } l > m$$

$$d = \frac{k}{m} \qquad \text{for } l \leq m$$

Therefore the dimensions for the matrices for the R.M.D. approach are:

$H$ is $lp$ x $d(l + m)$, where

$$d(l+m) = lp$$

$X$ is $d(l + m)$ x $m$, and $C$ is $lp$ x $m$, while the $A_i$ are $l$ x $m$ and the $B_i$ are $m$ x $m$. It is noticed that for the R.M.D. approach $H$ is a square matrix. However, when $l > m$ $H$ will not be square and no reduced Padé model is possible. Therefore, the latter case will involve the inversion of a larger dimension matrix with all the associated computational overheads.

Turning to the L.M.D. approach to the exact Padé method, the description of the simplified system is

$$\mathbf{R}(s) = \hat{D}_2^{-1}(s)\hat{N}_2(s)$$

where

$$\hat{N}_2(s) = \widetilde{A}_0 + \widetilde{A}_1 s + \widetilde{A}_2 s^2 + \ldots + \widetilde{A}_{d-1} s^{d-1}$$

$$\hat{D}_2(s) = \widetilde{B}_0 + \widetilde{B}_1 s + \widetilde{B}_2 s^2 + \ldots + \widetilde{B}_{d-1} s^{d-1} + I s^d$$

With the scalars $k$, $l$, $m$, $d$ and $p$ defined as above, the L.M.D. approach uses the first $p$ equations of the infinite set given by

$$\hat{N}_2(s) = \hat{D}_2(s)\left(C_0 + C_1 s + C_2 s^2 + \ldots\right)$$

which may be given as

$$
\begin{aligned}
\widetilde{A}_0^T &= C_0^T \widetilde{B}_0^T \\
\widetilde{A}_1^T &= C_1^T \widetilde{B}_0^T + C_0^T \widetilde{B}_1^T \\
&\vdots \\
\widetilde{A}_{\frac{k}{m}-1}^T &= C_{\frac{k}{m}-1}^T \widetilde{B}_0^T + C_{\frac{k}{m}-2}^T \widetilde{B}_1^T + \ldots + C_0^T B_{\frac{k}{m}-1}^T \\
\mathbf{0} &= C_{\frac{k}{m}}^T \widetilde{B}_0^T + C_{\frac{k}{m}-1}^T \widetilde{B}_1^T + \ldots + C_1^T \widetilde{B}_{\frac{k}{m}-1}^T + C_0^T \\
&\vdots \\
\mathbf{0} &= C_{p-1}^T \widetilde{B}_0^T + C_{p-2}^T \widetilde{B}_1^T + \ldots + C_{\frac{k}{l}}^T \widetilde{B}_{\frac{k}{m}-1}^T + C_{\frac{k}{l}-1}^T
\end{aligned}
\tag{7.8}
$$

The transposition of the matrices is required so that (7.8) can be written in the matrix form (7.7) with

$$
H = \begin{bmatrix}
I_l & \mathbf{0} & \cdots & \mathbf{0} & -C_0^T & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{0} & I_l & \cdots & \mathbf{0} & -C_1^T & -C_0^T & \cdots & \mathbf{0} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{0} & \mathbf{0} & \cdots & I_l & -C_{\frac{k}{m}-1}^T & -C_{\frac{k}{m}-2}^T & \cdots & -C_0^T \\
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -C_{\frac{k}{m}}^T & -C_{\frac{k}{m}-1}^T & \cdots & -C_1^T \\
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -C_{\frac{k}{m}+1}^T & -C_{\frac{k}{m}}^T & \cdots & -C_2^T \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -C_{p-1}^T & -C_{p-2}^T & \cdots & -C_{\frac{k}{l}}^T
\end{bmatrix}
$$

$$
X = \begin{bmatrix}
\widetilde{A}_0^T \\
\widetilde{A}_1^T \\
\vdots \\
\widetilde{A}_{d-1}^T \\
\widetilde{B}_0^T \\
\widetilde{B}_1^T \\
\vdots \\
\widetilde{B}_{d-1}^T
\end{bmatrix}
\qquad \text{and} \qquad
C = \begin{bmatrix}
\mathbf{0} \\
\mathbf{0} \\
\vdots \\
\mathbf{0} \\
C_0^T \\
C_1^T \\
\vdots \\
C_{\frac{k}{l}-1}^T
\end{bmatrix}
$$

Therefore the dimensions for the matrices for the L.M.D. approach are:

$H$ is $lm$ x $d(l+m)$, $X$ is $d(l+m)$ x $l$, and $C$ is $lm$ x $l$, while the $A_i$ are $l$ x $m$ and the $B_i$ are $l$ x $l$. In this case the matrix $H$ will be square when $l > m$.

Therefore, it is clear that there are two distinct approaches to implementing the exact Padé method in the multivariable case depending on whether the L.M.D. or R.M.D. is used for the reduced order transfer function matrix. In the case where $l = m$ there is no saving in calculation involved in choosing one approach over the other. However, in all other cases, the size of the matrix $H^T H$ to be inverted will be affected by the choice of matrix description employed. In particular, when $l < m$ the R.M.D. approach will involve the inversion of a matrix of smaller dimension, but if $l > m$ the L.M.D. approach will be more economical.

**The LS Padé Method**

The extension to LS Padé approximation is analogous to the SISO case in that, instead of using the information from the first $p$ time moment matrices to calculate a $k$th order model, the first $p + t$ time moment matrices are used ($t > 0$ is an integer). It is noted that the distinction between the exact and LS method is not so clear cut in the multivariable case because, as noted in the previous subsection, for certain orders no exact reduced model exists.

In other words, considering the R.M.D. approach, without loss of generality the first $p + t$ equations from the system given by

$$\hat{N}_1(s) = \left(C_0 + C_1 s + C_2 s^2 + \ldots\right)\hat{D}_1$$

are used to calculate the coefficients of the transfer function elements of the reduced model. This is the system of equations given in (7.6) extended as follows

$$
\begin{aligned}
A_0 &= C_0 B_0 \\
A_1 &= C_1 B_0 + C_0 B_1 \\
&\vdots \\
A_{\frac{k}{m}-1} &= C_{\frac{k}{m}-1} B_0 + C_{\frac{k}{m}-2} B_1 + \ldots + C_0 B_{\frac{k}{m}-1} \\
\mathbf{0} &= C_{\frac{k}{m}} B_0 + C_{\frac{k}{m}-1} B_1 + \ldots + C_1 B_{\frac{k}{m}-1} + C_0 \\
&\vdots \\
\mathbf{0} &= C_{p+t-1} B_0 + C_{p+t-2} B_1 + \ldots + C_{\frac{k}{l}+t} B_{\frac{k}{m}-1} + C_{\frac{k}{l}+t-1}
\end{aligned}
\tag{7.9}
$$

so that (7.7) is solved, in a least-squares sense, where

$$
H = \begin{bmatrix}
I_m & \mathbf{0} & \cdots & \mathbf{0} & -C_0 & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{0} & I_m & \cdots & \mathbf{0} & -C_1^T & -C_0 & \cdots & \mathbf{0} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{0} & \mathbf{0} & \cdots & I_m & -C_{k-1} & -C_{k-2} & \cdots & -C_0 \\
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -C_k & -C_{k-1} & \cdots & -C_1 \\
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -C_{k+1} & -C_k & \cdots & -C_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -C_{p+t-1} & -C_{p+t-2} & \cdots & -C_{p+t-k}
\end{bmatrix}
$$

$$
X = \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_{d-1} \\ B_0 \\ B_1 \\ \vdots \\ B_{d-1} \end{bmatrix} \qquad \textit{and} \qquad C = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ C_0 \\ C_1 \\ \vdots \\ C_{\frac{k}{l}+t-1} \end{bmatrix}
$$

This definition of the LS Padé method is given in terms of the R.M.D. of $\mathbf{R}(s)$, but clearly there exists a complementary definition in terms of the L.M.D. However, the question of

which approach will prove more attractive in a particular example depends on more than simply the issue of the size of the matrix $H^TH$ to be inverted.

In section 7.2 it was shown that the McMillan degree or the degree of the pole polynomial identified by consideration of the non-zero minors of all orders is the only reliable indicator of the order of the system represented by a given transfer function matrix. Therefore, the question of whether any actual order reduction is achieved by an LS Padé approximation depends on a comparison of the McMillan degrees of the full and simplified models. For this reason it is important to consider what factors will affect the degree of the pole polynomial of $\mathbf{R}(s)$.

Regardless of the matrix fraction description used for $\mathbf{R}(s)$ and provided we ignore the unlikely possibility of cancellations, the degree of the pole polynomial obtained by consideration of the non-zero minors of all orders of $\mathbf{R}(s)$ depends on two things. These are the degree, $d$, of the polynomial elements of the denominator matrix and the dimension of the denominator matrix.

Consider the case of an $l$ x $m$ system where $l < m$. For the R.M.D. approach we have seen that

$$\hat{D}_1(s) = B_0 + B_1 s + B_2 s^2 + \ldots + B_{d-1} s^{d-1} + I s^d$$

which has the same dimension $m$ x $m$ as the $B_i$. Each element of this denominator matrix will be a polynomial of degree $d$ which, assuming no cancellations, gives an inverse whose elements will have a common denominator of degree $dm$. Similarly, it is seen that for the L.M.D. approach the inverse of the denominator matrix will have a common denominator of degree $dl$. Since $l < m$ the McMillan degree of the simplified system produced by the L.M.D. approach in this case will almost certainly be less than that

produced by the R.M.D. approach and, hence, more likely to give actual order reduction. However, the R.M.D. approach involves the inversion of a matrix of dimension $l(p+t)$ x $l(p+t)$ at less computational cost than for the L.M.D. approach for which $H$ is $m(p+t)$ x $m(p+t)$. These issues are illustrated in the examples below.

**The Equivalence of Full and Partial Methods**

In the multivariable case, the matrix $H$ may be expressed in the partitioned form

$$H = \left[\begin{array}{c|c} I_\Delta & \mathbf{C}_0 \\ \hline \Phi & \mathbf{C}_1 \end{array}\right] \tag{7.10}$$

In this form, for the R.M.D. approach, $\Delta = dm$ and the elements of $\mathbf{C}_0$ and $\mathbf{C}_1$ are the time moment matrices from the power series expansion of $\mathbf{G}(s)$. Further, for the L.M.D. approach, $\Delta = dl$ and the elements of $\mathbf{C}_0$ and $\mathbf{C}_1$ are the transposes of the time moment matrices.

It is clear from (7.10) that for both approaches matrix equation (7.7) can be written in a partitioned form exactly analogous to the matrix-vector equation (5.3)

$$\left[\begin{array}{c|c} I_\Delta & \mathbf{C}_0 \\ \hline \Phi & \mathbf{C}_1 \end{array}\right]\left[\begin{array}{c} \mathbf{D} \\ \hline \mathbf{E} \end{array}\right] = \left[\begin{array}{c} \mathbf{0} \\ \hline \mathbf{Q} \end{array}\right] \tag{7.11}$$

where for the R.M.D. approach

$$\mathbf{D} = \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_{d-1} \end{bmatrix} \qquad \mathbf{E} = \begin{bmatrix} B_0 \\ B_1 \\ \vdots \\ B_{d-1} \end{bmatrix} \qquad \text{and} \qquad \mathbf{Q} = \begin{bmatrix} C_0 \\ C_1 \\ \vdots \\ C_{\frac{k}{l}+t-1} \end{bmatrix}$$

and for the L.M.D. approach

$$\mathbf{D} = \begin{bmatrix} \widetilde{A}_0^T \\ \widetilde{A}_1^T \\ \vdots \\ \widetilde{A}_{d-1}^T \end{bmatrix} \qquad \mathbf{E} = \begin{bmatrix} \widetilde{B}_0^T \\ \widetilde{B}_1^T \\ \vdots \\ \widetilde{B}_{d-1}^T \end{bmatrix} \qquad \text{and} \qquad \mathbf{Q} = \begin{bmatrix} C_0^T \\ C_1^T \\ \vdots \\ C_{\frac{k}{l}+t-1}^T \end{bmatrix}$$

The partitioned form (7.11) is such that the same analysis may be applied here as in section 5.2 to show that the LS solution

$$H^T HX = H^T C$$

is equivalent to the LS solution of the corresponding matrix equation for the partial method given by

$$\mathbf{C}_1^T \mathbf{C}_1 \mathbf{E} = \mathbf{C}_1^T \mathbf{Q}$$

followed by the matching of the first $k/m$ time moment matrices of $\mathbf{G}(s)$ and $\mathbf{R}(s)$.

**Example 7.2**

Consider the 2 input/2 output system represented by the transfer function matrix

$$\begin{bmatrix} \dfrac{8s^2+6s+2}{4s^3+13s^2+11s+2} & \dfrac{8s^2+6s+2}{9s^3+27s^2+20s+4} \\ \dfrac{8s^2+6s+2}{8s^3+30s^2+19s+3} & \dfrac{8s^2+6s+2}{8s^3+14s^2+7s+1} \end{bmatrix}$$

This system has a pole polynomial

$$p(s) = (4s+1)^2(3s+1)(2s+1)(3s+2)(s+1)^2(s+2)(s+3)$$

giving a McMillan degree of 9. If we take $k = 4$, on applying the LS Padé method using R.M.D., unstable models are obtained for $t = 0$ and $t = 1$. However, on increasing the number of extra time moment matrices used to 2 the following stable model is produced

$$\mathbf{R}(s) = \begin{bmatrix} \dfrac{0.388245s^3 + 0.317754s^2 + 0.0860854s + 0.00772606}{s^4 + 1.9959s^3 + 0.53539s^2 + 0.1054s + 0.00772606} & \dfrac{0.215586s^3 + 0.17196s^2 + 0.0449743s + 0.00386303}{s^4 + 1.9959s^3 + 0.53539s^2 + 0.1054s + 0.00772606} \\ \dfrac{0.206978s^3 + 0.181839s^2 + 0.0531015s + 0.00515096}{s^4 + 1.9959s^3 + 0.53539s^2 + 0.1054s + 0.00772606} & \dfrac{0.4971653s^3 + 0.4734125s^2 + 0.148992s + 0.0154521}{s^4 + 1.9959s^3 + 0.53539s^2 + 0.1054s + 0.00772606} \end{bmatrix}$$

The relative ISE results for the individual elements in $\mathbf{R}(s)$ are as follows

| | | |
|---|---|---|
| $R_{11}(s)$ | $I_{rel} = 59\%$ | $J_{rel} = 31\%$ |
| $R_{12}(s)$ | $I_{rel} = 49\%$ | $J_{rel} = 32\%$ |
| $R_{21}(s)$ | $I_{rel} = 49\%$ | $J_{rel} = 7.5\%$ |
| $R_{22}(s)$ | $I_{rel} = 12\%$ | $J_{rel} = 1.3\%$ |

The elements have a common denominator of degree 4 which is what would have been predicted from the foregoing analysis ($dm = 4$). The order of the system from examination of the non-zero minors of all orders is 8 because there is no cancellation. Actual order reduction has been achieved although not of a large amount and as in the SISO case a dramatic improvement in the results of the approximation achieved by the introduction of more system information into the calculation.

The same pattern of results is found for the L.M.D. approach when $k = 4$. Unstable models are produced for $t = 0$ and $t = 1$, but for $t = 2$ we obtain

$$\mathbf{R}(s) = \begin{bmatrix} \dfrac{0.374642s^3 + 0.316617s^2 + 0.084189s + 0.00714019}{s^4 + 1.16825s^3 + 0.510497s^2 + 0.0974477s + 0.00714019} & \dfrac{0.183126s^3 + 0.142028s^2 + 0.0436023s + 0.00476036}{s^4 + 1.16825s^3 + 0.510497s^2 + 0.0974477s + 0.00714019} \\ \dfrac{0.206201s^3 + 0.19834s^2 + 0.0543218s + 0.00357009}{s^4 + 1.16825s^3 + 0.510497s^2 + 0.0974477s + 0.00714019} & \dfrac{0.482536s^3 + 0.439201s^2 + 0.136575s + 0.0142803}{s^4 + 1.16825s^3 + 0.510497s^2 + 0.0974477s + 0.00714019} \end{bmatrix}$$

for the elements of which the relative ISE figures are

| | | |
|---|---|---|
| $R_{11}(s)$ | $I_{rel} = 61\%$ | $J_{rel} = 40.8\%$ |
| $R_{12}(s)$ | $I_{rel} = 59\%$ | $J_{rel} = 183\%$ |
| $R_{21}(s)$ | $I_{rel} = 51\%$ | $J_{rel} = 13\%$ |
| $R_{22}(s)$ | $I_{rel} = 13\%$ | $J_{rel} = 1.5\%$ |

For the same example and taking $k = 2$, the LS Padé method using the R.M.D. approach gives a stable reduced order model given by

$$\mathbf{R}(s) = \begin{bmatrix} \dfrac{0.23562s + 0.0533208}{s^2 + 0.461859s + 0.0533208} & \dfrac{0.129897s + 0.0266602}{s^2 + 0.461859s + 0.0533208} \\ \dfrac{0.155098s + 0.0355488}{s^2 + 0.461859s + 0.0533208} & \dfrac{0.462668s + 0.106641}{s^2 + 0.461859s + 0.0533208} \end{bmatrix}$$

with relative ISE results

| | | |
|---|---|---|
| $R_{11}(s)$ | $I_{rel} = 71\%$ | $J_{rel} = 63\%$ |
| $R_{12}(s)$ | $I_{rel} = 63.5\%$ | $J_{rel} = 77\%$ |
| $R_{21}(s)$ | $I_{rel} = 56\%$ | $J_{rel} = 13\%$ |
| $R_{22}(s)$ | $I_{rel} = 14\%$ | $J_{rel} = 1.6\%$ |

This choice of value for $k$ gives a reduced model with McMillan degree of 4 as opposed to the full system's McMillan degree of 9. Also, it is the greatest order reduction available using this method consistent with the conditions and relationships outlined in this section.

This example illustrates a number of points. Firstly, that an unstable result for the exact Padé method can be improved upon as in the SISO case. Secondly, that significant levels of order reduction are difficult to achieve. Thirdly, that the simplified models produced by the R.M.D. and L.M.D. approaches differ even in the case where $l = m$.

**Example 7.3**

We now consider the 3 input/2 output system represented by

$$\begin{bmatrix} \dfrac{8s^2+6s+2}{4s^3+13s^2+11s+2} & \dfrac{8s^2+6s+2}{9s^3+27s^2+20s+4} & \dfrac{8s^2+6s+2}{9s^3+27s^2+20s+4} \\ \dfrac{8s^2+6s+2}{8s^3+30s^2+19s+3} & \dfrac{8s^2+6s+2}{8s^3+14s^2+7s+1} & \dfrac{8s^2+6s+2}{4s^3+13s^2+11s+2} \end{bmatrix}$$

which has a McMillan degree of 10 from examination of the pole polynomial

$$p(s) = (4s+1)^2(3s+1)(2s+1)(3s+2)(s+1)^2(s+2)^2(s+3)$$

LS Padé approximation was applied to this system using the lowest value $k = 6$ which will return integer values for $k/l$ and $k/m$. For this example the R.M.D. approach gives unstable models for $t = 0$ and $t = 1$. However, the L.M.D. approach yields stable results for $t = 0$

$$\mathbf{R}(s) = \begin{bmatrix} \dfrac{0.3374s^3 + 0.189607s^2 + 0.0354541s + 0.00219679}{s^4 + 0.870252s^3 + 0.283821s^2 + 0.0409461s + 0.00219679} \\ \dfrac{0.190437s^3 + 0.113078s^2 + 0.022417s + 0.00146461}{s^4 + 0.870252s^3 + 0.283821s^2 + 0.0409461s + 0.00219679} \\ \dfrac{0.183642s^3 + 0.0991463s^2 + 0.01827625s + 0.0010984}{s^4 + 0.870252s^3 + 0.283821s^2 + 0.0409461s + 0.00219679} \\ \dfrac{0.483207s^3 + 0.305464s^2 + 0.0643179s + 0.0043936}{s^4 + 0.870252s^3 + 0.283821s^2 + 0.0409461s + 0.00219679} \\ \dfrac{0.15423s^3 + 0.0932519s^2 + 0.0182762s + 0.0010984}{s^4 + 0.870252s^3 + 0.283821s^2 + 0.0409461s + 0.00219679} \\ \dfrac{0.288427s^3 + 0.177409s^2 + 0.0354541s + 0.00219679}{s^4 + 0.870252s^3 + 0.283821s^2 + 0.0409461s + 0.00219679} \end{bmatrix}$$

and for $t = 1$

$$\mathbf{R}(s) = \begin{bmatrix} \dfrac{0.34979s^3 + 0.272188s^2 + 0.0691663s + 0.00573561}{s^4 + 1.1034s^3 + 0.455612s^2 + 0.0835052s + 0.00573561} \\ \dfrac{0.198206s^3 + 0.160161s^2 + 0.0429271s + 0.00382392}{s^4 + 1.1034s^3 + 0.455612s^2 + 0.0835052s + 0.00573561} \\ \dfrac{0.191465s^3 + 0.146232s^2 + 0.036017s + 0.00286781}{s^4 + 1.1034s^3 + 0.455612s^2 + 0.0835052s + 0.00573561} \\ \dfrac{0.491423s^3 + 0.42452s^2 + 0.121125s + 0.0114711}{s^4 + 1.1034s^3 + 0.455612s^2 + 0.0835052s + 0.00573561} \\ \dfrac{0.201485s^3 + 0.148416s^2 + 0.036017s + 0.00286781}{s^4 + 1.1034s^3 + 0.455612s^2 + 0.0835052s + 0.00573561} \\ \dfrac{0.373155s^3 + 0.278063s^2 + 0.0691662s + 0.00573561}{s^4 + 1.1034s^3 + 0.455612s^2 + 0.0835052s + 0.00573561} \end{bmatrix}$$

The relative ISE results for the model produced when $t = 1$ were

| | | |
|---|---|---|
| $R_{11}(s)$ | $I_{rel} = 62\%$ | $J_{rel} = 36\%$ |
| $R_{12}(s)$ | $I_{rel} = 53\%$ | $J_{rel} = 39\%$ |
| $R_{13}(s)$ | $I_{rel} = 51.5\%$ | $J_{rel} = 36\%$ |
| $R_{21}(s)$ | $I_{rel} = 50\%$ | $J_{rel} = 8\%$ |
| $R_{22}(s)$ | $I_{rel} = 12.5\%$ | $J_{rel} = 1.4\%$ |
| $R_{23}(s)$ | $I_{rel} = 60\%$ | $J_{rel} = 34\%$ |

It is interesting to note that the L.M.D. approach in this example not only produces stable reduced models but also that the models produced are of McMillan degree 8, whereas the unstable models produced by the R.M.D. approach are of McMillan degree 12. These advantages are gained at the cost of requiring the inversion of a 15 x 15 matrix for the L.M.D. approach rather than 10 x 10 for the R.M.D. approach. Clearly, for the LS Padé method applied to multivariable systems, these factors would need to be carefully considered when deciding on the preferred approach for a particular full system.

## 7.5 Remarks

In this chapter, the approach taken to the application of Padé approximation to the multivariable case has followed that of Shamash (1976). Matrix fraction descriptions of the full and reduced multivariable systems have been used to develop a clear outline of both Padé and LS-Padé approximation for this case. This work has highlighted a number of issues regarding the choice of R.M.D. or L.M.D. for performing the process in particular examples and, importantly, concerning the difficulty mentioned by Taiwo and Krebs (1993) of ensuring that actual reduction of the full system is achieved.

It is important to note that a number of authors have taken a mixed modal-Padé approach to this problem (Shieh and Wei 1975, Shamash 1975c, Bandyopadhyay and Lamba 1987 and Aguirre and Mendes 1995). This approach has been seen as avoiding certain complications arising for the Padé methods in the multivariable case. For example, Bandyopadhyay and Lamba (1987) produced important work on Padé approximation while proposing a modal-Padé method to overcome the fact that the $r$th order Padé approximant matches fewer than $r$ time moments in the multivariable case. It is noted that, as in the SISO case, the extension to an LS-Padé approximation by the inclusion of further time moment matrices could be proposed as an alternative answer to this problem.

Also recently, Aguirre and Mendes (1995) proposed a method which combines the stability preserving method of pole retention with the work of Aguirre (1995) on the LS approximation of numerator coefficients in the SISO case (section 4.5). This introduction of an LS extension to the modal-Padé approximation of multivariable systems is worthy of note. However, the authors acknowledge that for the method the final simplified models may not have a reduced McMillan degree compared with the original system.

# CHAPTER 8

# CONCLUSIONS

## 8.1 Results from the Thesis

A detailed survey of the literature on the subject of model reduction was carried out and is given in Chapters 2 – 4. From observations the author made carrying out that survey original insights were gained into exactly how the behaviour of a high order linear dynamical system is approximated by least-squares methods based upon the classic Padé method of model reduction. The conclusions drawn from this work may be summarized under three headings.

- Development of a framework for least-squares methods
- Proof of a stability preserving property in the discrete-time case
- Development of new least-squares methods

**A Framework for Least-squares Methods**

The literature survey revealed a situation in which a number of authors had proposed frequency-domain model reduction methods involving least-squares approximation. However, since the focus for much of this work was the practical consideration of simplifying control system design, there had been little examination of exactly how these methods approximated the full system or of the relationship between the various methods proposed. For this reason, the author adopted as one of the main objectives of the present work the development of a thorough and well-founded mathematical understanding of least-squares Padé methods.

Pursuit of this goal resulted in a two-stage analysis which proved to be an invaluable tool for realizing this and the other objectives of the research project. Indeed, the fundamental result upon which most of the rest of the project depends is the proof of the equivalence of the full and partial approaches to least-squares moment matching given in section 5.2 of the thesis. By partitioning the matrices in the matrix-vector equation

$$A\mathbf{x} = \mathbf{b}$$

in an appropriate fashion it became a simple matter to show that the least-squares solution of this equation to find the numerator *and* denominator coefficients of a reduced order model is equivalent to using least-squares to find the denominator coefficients only followed by exact matching of the first $k$ time moments for the numerator.

This basic result not only made clear possible computational savings in the application of least-squares approximation, but also gave a deeper understanding of how the method approximated the full system and enabled the clear formulation of the error index minimized. This analysis of least-squares moment matching could be seen immediately to lead to a nonuniqueness property of the method which, unlike the exact Padé method, can produce a variety of reduced models depending on which coefficient is set to unity.

The extension of this analysis to generalised least-squares Padé approximation obtained the result presented in section 5.5 that all the least-squares methods proposed so far in the literature were seen to be special cases within the general framework of the two-stage analysis. This divided the procedure into a least-squares approximation of the denominator coefficients followed by some form of matching or averaging system

parameters for the numerator. Therefore, the understanding gained in the moment matching, continuous-time case, including the expression for the relevant error index, has been generalised. Hence, the research resulted in the clear statement of the relationship between previously proposed least-squares methods being made possible. Similarly, for the first time the nature of the errors minimised in the least-squares approximation could be stated clearly.

With this relationship clarified, the flexibility and adaptability of this approach is starkly highlighted. The further example given at the end of section 5.5 demonstrates the improved accuracy that may be achieved in particular examples by exploiting fully the simplicity and flexibility of the method. The clear superiority, in this example, of the least-squares result over the other methods, combined with the computational overheads incurred by using Routh approximation or the Stability Equation method, brings out clearly the attraction offered by least-squares Padé approximation.

A restriction on the generalisation of the two-stage analysis was noted in section 5.6 where consideration was given to the introduction of weighting to improve performance. When the use of distinct arbitrary weights was investigated it became clear that one of the matrices involved in the analysis of the relationship between the full and partial methods had to be of particular form, namely,

$$\left[\begin{array}{c|c} f_1 I_r & \Phi \\ \hline \Phi & f_2 I_{k-r} \end{array}\right]$$

before the full least-squares method involved could be fitted into the general framework developed. In the case of the weighted least-squares method with arbitrary distinct weights the relevant matrix was seen *not* to be of the necessary form.

**Stability Preserving in Discrete-Time**

Observation of the results of the model reduction of many discrete-time systems using Markov parameter matching only produced a *prima facia* case for believing that this was stability preserving. This empirical-based conviction was converted to one of mathematical certainty with the development of the proof given of this result in section 6.2 of the thesis. This proof puts the technique of least-squares Padé approximation on a firm theoretical foundation, showing at the same time how stability preservation breaks down if any other than the highest power denominator coefficient is set to unity.

This work and the initial steps in developing the two-stage analysis was provoked by the previous work of Lalonde (1992a, 1992b) on the combining of system identification with model reduction for the simplification of high order, non-linear systems. Indeed, since least-squares methods require only system parameter information which may be gathered experimentally, they are *not* restricted in their application to linear systems and may be applied without incurring the overhead of characterizing the original system analytically.

In the section 6.4 of the same chapter it is also clearly shown how the two-stage analysis of the method breaks down for the case of generalised least-squares Padé approximation in the discrete-time case. It is demonstrated there how it fails for exactly the reason that the crucial matrix in the development of the analysis is not of the required form.

**New Least-squares Methods**

The most significant new method proposed in the thesis is a new stability preserving method for the continuous-time case which is described in section 6.3 and is based on the stability preserving property proved for the discrete-time case. It utilizes a bilinear transformation from the $s$-plane to the $z$-plane and its inverse to reduce the continuous-time model reduction calculation to a discrete-time calculation that may be carried out making use of the stability guarantee given by the above-mentioned property. Results are shown to be as good as existing stability preservation methods.

Other new methods outlined are the rather disappointing 'optimal' least-squares method described in section 5.4 and the least-squares Padé method for multivariable systems demonstrated in Chapter 7. The issues raised by the application of both of these techniques were explored.

**Further Work**

The application of the least-squares Padé method to the multivariable case raises several important questions that could be answered fully given further research effort in this area. These questions are

- Can some reliable method of predicting the McMillan degree of the simplified model be identified?
- How computationally expensive would such a method be?
- How useful are existing SISO approximation error indices for assessing performance in the multivariable case?
- Can new, more meaningful, indices be defined?

Such questions and the general complexity of the multivariable case in the frequency domain suggest that this is an area rich in research possibilities.

# APPENDIX 1

In section 6.2 the value of $p_j$ which minimises

$$\sum_{n=0}^{\infty}\left|\varepsilon_n^2\right|$$

in (6.8) is used in the proof of a stability preservation property of the LS Padé method for discrete-time systems using Markov parameters only. A derivation of the expression for this value of $p_j$ is now given.

If we assume the equation (6.8)

$$u_{n+1} - p_j u_n = \varepsilon_n$$

then

$$\left|\varepsilon_n^2\right| = \left(\hat{u}_{n+1} - \hat{p}_j \hat{u}_n\right)\left(u_{n+1} - p_j u_n\right)$$

where ($\hat{a}$ denotes the complex conjugate of $a$)

$$p_j = \alpha + i\beta \text{ and } \hat{p}_j = \alpha - i\beta$$

giving

$$\left|\varepsilon_n^2\right| = \left[\hat{u}_{n+1} - (\alpha - i\beta)\hat{u}_n\right]\left[u_{n+1} - (\alpha + i\beta)u_n\right]$$

Summing these error terms for $n = 0$ to $\infty$ gives the following error index

$$E = \sum_{n=0}^{\infty}\left|\varepsilon_n^2\right| = \sum_{n=0}^{\infty}\left[\hat{u}_{n+1} - (\alpha - i\beta)\hat{u}_n\right]\left[u_{n+1} - (\alpha + i\beta)u_n\right]$$

and the condition for the minimising of $E$ is given by the partial derivatives of $E$ with respect to $\alpha$ and $\beta$ being equal to zero, i.e.,

$$\frac{\partial E}{\partial \alpha} = \frac{\partial E}{\partial \beta} = 0$$

Therefore, to derive an expression for the value of $p_j$ which minimises $E$ it is necessary to partially differentiate the sum, term by term, with respect to $\alpha$ and $\beta$ respectively. This is a simple process when it is seen that it simply involves summing the expressions for the derivatives of the $n$th term given by

$$\left|\varepsilon_n^2\right| = [\hat{u}_{n+1} - (\alpha - i\beta)\hat{u}_n][u_{n+1} - (\alpha + i\beta)u_n]$$

Therefore

$$\frac{\partial E}{\partial \alpha} = \sum_{n=0}^{\infty} \{-\hat{u}_n[u_{n+1} - (\alpha + i\beta)u_n] - u_n[\hat{u}_{n+1} - (\alpha - i\beta)\hat{u}_n]\}$$

$$= -2\sum_{n=0}^{\infty} \mathrm{Re}(\hat{u}_n u_{n+1} - \hat{u}_n p_j u_n)$$

and

$$\frac{\partial E}{\partial \beta} = \sum_{n=0}^{\infty} \{i\hat{u}_n[u_{n+1} - (\alpha + i\beta)u_n] - iu_n[\hat{u}_{n+1} - (\alpha - i\beta)\hat{u}_n]\}$$

$$= 2\sum_{i=0}^{\infty} i\,\mathrm{Im}(\hat{u}_n u_{n+1} - \hat{u}_n p_j u_n)$$

Hence, from the minimisation condition we obtain the following pair of equations

$$\sum_{n=0}^{\infty} \mathrm{Re}(\hat{u}_{ni} u_{n+1} - \hat{u}_n p_j u_n) = 0$$

$$\sum_{i=0}^{\infty} i\,\mathrm{Im}(\hat{u}_n u_{n+1} - \hat{u}_n p_j u_n) = 0$$

which on addition gives

$$\sum_{n=0}^{\infty}\mathrm{Re}\left(\hat{u}_n u_{n+1}-\hat{u}_n p_j u_{n+1}\right)+i\sum_{n=0}^{\infty}\mathrm{Im}\left(\hat{u}_n u_{n+1}-\hat{u}_n p_j u_{n+1}\right)=0$$

$$\Rightarrow \sum_{n=0}^{\infty}\left(\hat{u}_n u_{n+1}-\hat{u}_n p_j u_n\right)=0$$

$$\Rightarrow p_j\sum_{n=0}^{\infty}\hat{u}_n u_n=\sum_{n=0}^{\infty}\hat{u}_n u_{n+1}$$

$$\Rightarrow p_j=\frac{\sum_{n=0}^{\infty}\hat{u}_n u_{n+1}}{\sum_{n=0}^{\infty}\hat{u}_n u_n} \qquad \text{(A1.1)}$$

Hence, the expression given in (A1.1) gives the value of $p_j$ which minimises the error index $E$.

## APPENDIX 2

Consider the identity,

$$\left(|u_1| - |u_2|\right)^2 \geq 0$$

where $u_1$ and $u_2$ are complex numbers and the equality sign holds when

$$|u_1| = |u_2|$$

This leads to the inequality

$$|u_1|^2 + |u_2|^2 \geq 2|u_1||u_2| \tag{A2.1}$$

Similarly, for another complex number $u_3$ the following inequalities will hold

$$|u_2|^2 + |u_3|^2 \geq 2|u_2||u_3| \tag{A2.2}$$

$$|u_3|^2 + |u_1|^2 \geq 2|u_3||u_1| \tag{A2.3}$$

If we add (A2.1), (A2.2) and (A2.3) we obtain the further inequality

$$|u_1|^2 + |u_2|^2 + |u_3|^2 \geq |u_1||u_2| + |u_2||u_3| + |u_3||u_1|$$

and, continuing in this fashion, it follows that

$$\sum_{i=1}^{n} |u_i|^2 \geq \sum_{i=1}^{n-1} |u_i||u_{i+1}| + |u_n||u_1| \tag{A2.4}$$

If we assume that $|u_i| \to 0$ as $i \to \infty$, then taking the limit as $n \to \infty$ in (A2.4) gives

$$\sum_{i=1}^{\infty}|u_i|^2 \geq \sum_{i=1}^{\infty}|u_i||u_{i+1}|$$

Now for such a sequence of complex numbers $\{u_i\}$, all of its values cannot be equal (unless they are all identically zero) and so the strict inequality holds

$$\sum_{i=1}^{\infty}|u_i|^2 > \sum_{i=1}^{\infty}|u_i||u_{i+1}| \qquad \text{(A2.5)}$$

giving the result

$$\frac{\sum_{i=1}^{\infty}|u_i||u_{i+1}|}{\sum_{i=1}^{\infty}|u_i|^2} < 1$$

used in section 6.2.

# APPENDIX 3

```
100 CLEAR , , 3072
    COLOR 0, 10: CLS
    DEFINT I-K, M-N
    DEFDBL A-G, P-T, W-X, Z
    H$ = "###.#####"
    LOCATE 5, 5
    PRINT "P R O G R A M   T O   P R O D U C E   R E D U C E D   M O D E L S   O
    LOCATE CSRLIN + 1, 8
    PRINT "D I S C R E T E   T I M E   S Y S T E M S   U S I N G   L E A S T - S
    LOCATE CSRLIN + 1, 15
    PRINT "  M E T H O D S   I N V O L V I N G   M A R K O V   P A R A M E T E R
    LOCATE CSRLIN + 2, 22
    PRINT "T. N.  L U C A S  &  I. D.  S M I T H"
    LOCATE CSRLIN + 4, 10
    INPUT "Order of full system is - ", N
    NSYS = N
    NZ = 0
    DIM GN(N), GD(N), GN1(N)
190 PRINT : PRINT
    PRINT "   Enter the numerator coefficients from lowest powers"
    PRINT
    SN = 0
    FOR I = 0 TO N

        INPUT ; "    ", GN(I)
        SN = SN + GN(I): GN1(I) = GN(I)
    NEXT I
    PRINT
    PRINT : PRINT "   Enter the denominator coefficients from lowest powers"
    PRINT
    SD = 0
    FOR I = 0 TO N
        INPUT ; "    ", GD(I)
        SD = SD + GD(I)
    NEXT I
    CLS
    LOCATE CSRLIN + 2, 10
    PRINT "Transfer Function entered is :"
    LOCATE CSRLIN + 2, 10
    FOR I = 0 TO N
        IF GN(I) = 0 GOTO 450
        IF POS(C) >= 65 THEN LOCATE CSRLIN + 2, 10
        IF GN(I) < 0 THEN PRINT " - ";  ELSE IF I <> 0 THEN PRINT " + ";
        IF ABS(GN(I)) = 1! AND I <> 0 GOTO 410
        PRINT CSNG(ABS(GN(I)));
410     IF I > 0 THEN PRINT "z";
        LOCATE CSRLIN - 1, POS(C)
        IF I > 1 THEN PRINT I;
        LOCATE CSRLIN + 1, POS(C)
450 NEXT I
    NPOS = POS(C): LOCATE CSRLIN + 1, 10
    IF NPOS < 9 * N + 10 THEN NPOS = 9 * N + 10
    FOR J = 10 TO NPOS
        PRINT "—";
    NEXT J
    LOCATE CSRLIN + 2, 10
    FOR I = 0 TO N
        IF GD(I) = 0 GOTO 620
        IF POS(C) >= 65 THEN LOCATE CSRLIN + 2, 10
        IF GD(I) < 0 THEN PRINT " - ";  ELSE IF I <> 0 THEN PRINT " + ";
        IF ABS(GD(I)) = 1! AND I <> 0 GOTO 580
        PRINT CSNG(ABS(GD(I)));
580     IF I > 0 THEN PRINT "z";
```

```
    NEXT I
    IF K1 = 0 THEN 1100

    REM ***** PRE-MULTIPLY PARAMETER VECTOR BY TRANSPOSE ******

    FOR I = 1 TO 2 * K + 1
        T1 = 0
        FOR J = 1 TO 2 * K + K1 + 1
                T1 = T1 + AP(I, J) * D1(J - 1)
        NEXT J
        AQ2(I) = T1
    NEXT I

    ERASE D1

1100 REM ***** INVERSION ROUTINE ******

    FOR I = 1 TO 2 * K
        JV = I
        FOR J = I + 1 TO 2 * K + 1
                IF ABS(AUG(JV, I)) < ABS(AUG(J, I)) THEN JV = J
        NEXT J
        IF JV <> I THEN
                FOR JC = 1 TO 4 * K + 2
                        TM = AUG(I, JC)
                        AUG(I, JC) = AUG(JV, JC)
                        AUG(JV, JC) = TM
                NEXT JC
        END IF
        IF AUG(I, I) = 0 THEN GOTO 1200
        FOR JR = I + 1 TO 2 * K + 1
                IF AUG(JR, I) <> 0 THEN
                        R = AUG(JR, I) / AUG(I, I)
                        FOR KC = I TO 4 * K + 2
                                TEMP = AUG(JR, KC)
                                AUG(JR, KC) = AUG(JR, KC) - R * AUG(I, KC)
                                IF ABS(AUG(JR, KC)) < .000000001# THEN AUG(JR,
                        NEXT KC
                END IF
        NEXT JR
    NEXT I


    IF AUG(2 * K + 1, 2 * K + 1) = 0 THEN GOTO 1200
    FOR M = 2 * K + 2 TO 4 * K + 2
        AUG(2 * K + 1, M) = AUG(2 * K + 1, M) / AUG(2 * K + 1, 2 * K + 1)
        FOR NV = 2 * K TO 1 STEP -1
                TA = AUG(NV, M)
                FOR KV = NV + 1 TO 2 * K + 1
                        TA = TA - AUG(NV, KV) * AUG(KV, M)
                NEXT KV
                AUG(NV, M) = TA / AUG(NV, NV)
        NEXT NV
    NEXT M
    IF K1 = 0 THEN
        FOR I = 1 TO 2 * K + 1
                FOR J = 2 * K + 2 TO 4 * K + 2
                        M = J - 2 * K - 1
                        AQ1(I, M) = AUG(I, J)
                NEXT J
        NEXT I
        FOR I = 1 TO 2 * K + 1
                T1 = 0
                FOR J = 1 TO 2 * K + 1
```

```
                        T1 = T1 + AQ1(I, J) * D1(J - 1)
                NEXT J
                AQ3(I) = T1
        NEXT I
        ERASE D1
        GOTO 1400
    END IF

    GOTO 1300

1200 PRINT "Matrix is singular!"
     GOTO 3710

1300 REM ***** PRE-MULTIPLY INTERMEDIATE VECTOR BY INVERSE *****

    FOR I = 1 TO 2 * K + 1
        FOR J = 2 * K + 2 TO 4 * K + 2
                M = J - 2 * K - 1
                AQ1(I, M) = AUG(I, J)
        NEXT J
    NEXT I


    FOR I = 1 TO 2 * K + 1
        T1 = 0
        FOR J = 1 TO 2 * K + 1
                T1 = T1 + AQ1(I, J) * AQ2(J)
        NEXT J
        AQ3(I) = T1
    NEXT I

1400 DIM RN(K), RD(K)
    FOR I = 0 TO K
        RN(I) = AQ3(I + 1)
    NEXT I
    FOR I = 0 TO K - 1
        RD(I) = AQ3(K + I + 2)
    NEXT I
    RD(K) = 1


    IF M$ = "S" OR M$ = "s" THEN
        IF GN1(K) <> 0 THEN RN(K) = RN(K - 1) + G1
        FOR I = K - 1 TO 1 STEP -1
                RN(I) = RN(I - 1) + G1 * RD(I) - RN(I)
        NEXT I
        RN(0) = G1 * RD(0) - RN(0)
    END IF
1500 ERASE AH, AP, ID, AQ, AQ1, AQ2, AQ3, AUG
    GOTO 3100


1550 REM ***** CALCULATIONS TO OBTAIN DENOMINATOR ONLY ******

    DIM AH(K + K1, K), AP(K, K + K1)
    DIM ID(K, K), AQ(K, K), AQ3(K)
    DIM AQ1(K, K), AQ2(K), AUG(K, 2 * K)

    GOSUB 4200

    ERASE A, AX, BX, C, D

    FOR I = 1 TO K + K1
        FOR J = 1 TO K
```

```
                L = I + J
                AH(I, J) = -D1(L - 1)
            NEXT J
        NEXT I
        IF K1 = 0 THEN
            FOR I = 1 TO K
                FOR J = 1 TO K
                        AQ(I, J) = AH(I, J)
                NEXT J
            NEXT I
            GOTO 1600
        END IF

        REM ***** TRANSPOSING MATRIX H *****

        FOR I = 1 TO K
            FOR J = 1 TO K + K1
                AP(I, J) = AH(J, I)
            NEXT J
        NEXT I

        REM ***** PRE-MULTIPLYING H BY THE TRANSPOSE *****

        FOR I = 1 TO K
            FOR M = 1 TO K
                T1 = 0
                FOR J = 1 TO K + K1
                T1 = T1 + AP(I, J) * AH(J, M)
                NEXT J
                AQ(I, M) = T1
            NEXT M
        NEXT I

1600    FOR I = 1 TO K
            FOR J = 1 TO K
                AUG(I, J) = AQ(I, J)
            NEXT J
        NEXT I

        FOR I = 1 TO K
            FOR J = K + 1 TO 2 * K
                IF I = J - K THEN
                        AUG(I, J) = 1
                ELSE
                        AUG(I, J) = 0
                END IF
            NEXT J
        NEXT I
        IF K1 = 0 THEN 1650

        REM ***** PRE-MULTIPLY PARAMETER VECTOR BY TRANSPOSE ******

        FOR I = 1 TO K
            T1 = 0
            FOR J = 1 TO K + K1
                T1 = T1 + AP(I, J) * D1(J + K)
            NEXT J
            AQ2(I) = T1
        NEXT I

1650    REM ***** INVERSION ROUTINE ******

        FOR I = 1 TO K - 1
            JV = I
```

```
        FOR J = I + 1 TO K
                IF ABS(AUG(JV, I)) < ABS(AUG(J, I)) THEN JV = J
        NEXT J
        IF JV <> I THEN
                FOR JC = 1 TO 2 * K
                        TM = AUG(I, JC)
                        AUG(I, JC) = AUG(JV, JC)
                        AUG(JV, JC) = TM
                NEXT JC
        END IF
        IF AUG(I, I) = 0 THEN GOTO 1700
        FOR JR = I + 1 TO K
                IF AUG(JR, I) <> 0 THEN
                        R = AUG(JR, I) / AUG(I, I)
                        FOR KC = I TO 2 * K
                                TEMP = AUG(JR, KC)
                                AUG(JR, KC) = AUG(JR, KC) - R * AUG(I, KC)
                                IF ABS(AUG(JR, KC)) < .0000001 THEN AUG(JR, KC
                        NEXT KC
                END IF
        NEXT JR
    NEXT I


    IF AUG(K, K) = 0 THEN GOTO 1700
    FOR M = K + 1 TO 2 * K
        AUG(K, M) = AUG(K, M) / AUG(K, K)
        FOR NV = K - 1 TO 1 STEP -1
                TA = AUG(NV, M)
                FOR KV = NV + 1 TO K
                        TA = TA - AUG(NV, KV) * AUG(KV, M)
                NEXT KV
                AUG(NV, M) = TA / AUG(NV, NV)
        NEXT NV
    NEXT M
    IF K1 = 0 THEN
        FOR I = 1 TO K
                FOR J = K + 1 TO 2 * K
                        M = J - K
                        AQ1(I, M) = AUG(I, J)
                NEXT J
        NEXT I
        FOR I = 1 TO K
                T1 = 0
                FOR J = 1 TO K
                        T1 = T1 + AQ1(I, J) * D1(J + K)
                NEXT J
                AQ3(I) = T1
        NEXT I
        GOTO 1760
    END IF
    GOTO 1750

1700 PRINT "Matrix is singular!"
     GOTO 3710

1750 REM ***** PRE-MULTIPLY INTERMEDIATE VECTOR BY INVERSE *****

    FOR I = 1 TO K
        FOR J = K + 1 TO 2 * K
                M = J - K
                AQ1(I, M) = AUG(I, J)
        NEXT J
    NEXT I
```

```
    FOR I = 1 TO K
        T1 = 0
        FOR J = 1 TO K
                T1 = T1 + AQ1(I, J) * AQ2(J)
        NEXT J
        AQ3(I) = T1
    NEXT I

1760 DIM RN(K), RD(K)


    FOR I = 0 TO K - 1
        RD(I) = AQ3(I + 1)
    NEXT I
    RD(K) = 1

    FOR I = 0 TO K
        T1 = 0
        FOR J = 0 TO K - I
                T1 = T1 + D1(J) * RD(J + I)
        NEXT J
        RN(I) = T1
    NEXT I
    IF M$ = "S" OR M$ = "s" THEN
        IF GN1(K) <> 0 THEN RN(K) = RN(K - 1) + G1
        FOR I = K - 1 TO 1 STEP -1
                RN(I) = RN(I - 1) + G1 * RD(I) - RN(I)
        NEXT I
        RN(0) = G1 * RD(0) - RN(0)
    END IF


1800 ERASE AH, AP, ID, AQ, AQ1, AQ2, AQ3, AUG, D1


3100 REM ****** Routine to pass reduced system coefficients to SSE routine ***

    DIM X(2 * K + 2)
    IF M$ = "I" OR M$ = "i" THEN
        FOR I = 1 TO K
        IF CON = 0 THEN
                X(I) = RN(K - I)
                ELSEIF I = 1 THEN
                        X(1) = RN(K - 1) - CON * RD(K - 1)
                ELSE X(I) = RN(K - I) - CON * RD(K - I)
        END IF
        NEXT I
        ELSE
                L = 0
                FOR I = 1 TO K
                        A = 0: B = 0
                        FOR J = 0 TO L
                        A = A + RD(K - J)
                        B = B + RN(K - J)
                        NEXT J
                        L = L + 1
                        X(I) = B - G1 * A
                NEXT I
        END IF
        FOR I = 1 TO K
                X(K + I) = RD(K - I)
        NEXT I
```

```
     REM ****** Calculation of impulse/step energy of full system ******

3130 IF NZ = 1 THEN 3300
     N = NSYS
     DIM RDEN(2 * N + 2, N + 1), RNUM(2 * N + 2, N + 1)
     FOR I = 1 TO N + 1
         RDEN(1, I) = GD(I - 1): RDEN(2, I) = GD(N + 1 - I)
         IF M$ = "s" OR M$ = "S" THEN RNUM(1, I) = GNS(I - 1) ELSE RNUM(1, I)
     NEXT I
     NSTAB = 1
     GOSUB 3850
     Q1 = Q
     NZ = 1
     ERASE RDEN, RNUM

     REM - Formation of G(z)-G*(z) for SSE -

3300 N1 = NSYS + K + 1
     DIM RD1(N1), RD2(N1), RN1(N1), RN2(N1)
     FOR I = 1 TO NSYS + 1
         IF I <= K THEN RN2(I) = X(K + 1 - I)
         IF I <= K + 1 THEN RD2(I) = X(2 * K + 1 - I)
         IF M$ = "i" OR M$ = "I" THEN 3380
         IF I <= NSYS THEN RN1(I) = GNS(I - 1)
         GOTO 3390
3380     IF I <= NSYS THEN RN1(I) = GN(I - 1)
3390     IF I <= NSYS + 1 THEN RD1(I) = GD(I - 1)
     NEXT I
     RD2(K + 1) = 1
     DIM AN(N1), BM(N1)
     FOR I = 1 TO N1
         W1 = 0: W2 = 0
         FOR J = 1 TO I
             W1 = W1 + RD1(J) * RD2(I - J + 1)
             W2 = W2 + RN1(J) * RD2(I - J + 1) - RD1(J) * RN2(I - J + 1)
         NEXT J
         AN(I) = W2: BM(I) = W1
     NEXT I
     ERASE RD1, RD2, RN1, RN2
     DIM RDEN(2 * N1, N1), RNUM(2 * N1, N1)
     FOR I = 1 TO N1
         RDEN(1, I) = BM(I): RDEN(2, I) = BM(N1 - I + 1)
         RNUM(1, I) = AN(I)
     NEXT I
     N = NSYS + K
     NSTAB = 2
     GOSUB 3850
3400 Q3 = Q
     PRINT
     PRINT "          Reduced numerator (from highest powers)"
     PRINT "          ***************************************"
     PRINT
     FOR I = K TO 0 STEP -1
        PRINT "    ";
        PRINT USING H$; RN(I);
     NEXT I
     PRINT : PRINT
     PRINT "          Reduced denominator"
     PRINT "          *******************"
     PRINT
     FOR I = K TO 0 STEP -1
```

```
         PRINT "     ";
         PRINT USING H$; RD(I);
      NEXT I
      PRINT
      FOR J = 1 TO 80
         PRINT "-";
      NEXT J
      PRINT
      IF Q1 = 0 THEN 3660
      IF M$ = "s" OR M$ = "S" THEN 3650
      PRINT "    Relative Impulse SSE = "; CSNG(100 * Q3 / Q1); "%  :   Impulse
      GOTO 3660
3650  PRINT "    Relative Step SSE = "; CSNG(100 * Q3 / Q1); "%  :   Step SSE =
3660  ERASE AN, BM, RDEN, RNUM
      FOR J = 1 TO 80
          PRINT "-";
      NEXT J: PRINT
3700  PRINT
3710  PRINT
      INPUT "      Do you want a printout of the results? (y/n) - ", Y1$
      IF Y1$ = "y" OR Y1$ = "Y" THEN
         IF M$ = "i" OR M$ = "I" THEN RES$ = " IMPULSE" ELSE RES$ = "  STEP"
         IF M1$ = "f" OR M1$ = "F" THEN MD$ = " SMITH" ELSE MD$ = " MUNRO"
         LPRINT MD$; "  MARKOV:"; K1; RES$; " REL. SSE:"; CSNG(100 * Q3 / Q1);
         LPRINT : LPRINT
         FOR I = K TO 0 STEP -1
                 IF I = 6 THEN LPRINT
                 LPRINT "     ";
                 LPRINT USING H$; RN(I);
         NEXT I
         LPRINT : LPRINT
         FOR I = K TO 0 STEP -1
                 IF I = 6 THEN LPRINT
                 LPRINT "     ";
                 LPRINT USING H$; RD(I);
         NEXT I
         LPRINT
         FOR J = 1 TO 80
                 LPRINT "-";
         NEXT J
         LPRINT
      END IF

      INPUT ; "      Do you want to enter a different reduced model? (y/n) - ",
      IF (Y$ = "n") OR (Y$ = "N") THEN 3780
      ERASE X, RD, RN
      N = NSYS
      CLS
      GOTO 730
3780  PRINT
      INPUT "      Do you want to enter a different full system? (y/n) - ", U$
      IF (U$ = "n") OR (U$ = "N") THEN 4110
      ERASE X, RD, RN, GD, GN
      CLS
      N = NSYS
      GOTO 100

3850  REM - Alpha/Beta subroutine -

      Q = RNUM(1, 1) ^ 2 / RDEN(2, 1)
      FOR I = 3 TO 2 * N + 1 STEP 2
          FOR J = 1 TO N + 1 - INT(I / 2)
              RDEN(I, J) = RDEN(I - 2, J + 1) - RDEN(I - 2, 1) * RDEN(I - 1, J
              RDEN(I + 1, N + 2 - J - INT(I / 2)) = RDEN(I, J)
```

```
            RNUM(I, J) = RNUM(I - 2, J + 1) - RNUM(I - 2, 1) * RDEN(I - 1, J
        NEXT J
        IF RDEN(I + 1, 1) <= 0! THEN 3990
        Q = Q + RNUM(I, 1) ^ 2 / RDEN(I + 1, 1)
    NEXT I
    Q = Q / RDEN(2, 1)
    RETURN
3990 PRINT
    IF NSTAB = 1 THEN
       PRINT "   System is UNSTABLE"
       ELSE
       PRINT "   Model is UNSTABLE !!!"
    END IF
    ERASE RNUM, RDEN, AN, BM
    GOTO 3400
    PRINT
4110 PRINT : PRINT
    END

4200 REM ***** CALCULATION OF MARKOV PARAMETERS *****


    IF M$ = "i" OR M$ = "I" THEN
        FOR I = 0 TO N
                A(I) = GN1(I)
        NEXT I
    ELSE
        FOR I = 0 TO N
                A(I) = GNS(I)
        NEXT I
    END IF
    T3 = 0
    FOR I = 0 TO N - 1
        IF A(N) <> 0 THEN
                GOTO 4250
        ELSE
                FOR J = N TO 1 STEP -1
                        A(J) = A(J - 1)
                NEXT J
                A(0) = 0
                T3 = T3 + 1
        END IF
    NEXT I

4250 FOR I = 0 TO N
        BX(I) = GD(N - I)
        AX(I) = A(N - I)
    NEXT I
    K2 = 200
    FOR I = 0 TO N
        C(I) = AX(I)
    NEXT I
    D(0) = C(0) / BX(0)
    FOR I = 1 TO K2
        FOR J = 0 TO N
                C(J) = C(J + 1) - BX(J + 1) * D(I - 1)
        NEXT J
        D(I) = C(0) / BX(0)
    NEXT I

    FOR I = T3 TO K2 - T3
        D1(I) = D(I - T3)
    NEXT I
    RETURN
```

## References


Aguirre, L.A. 1992 The least-squares Padé method for model reduction. *Int. J. Systems Sci.*, **23:** 1559-1570.

Aguirre, L.A. 1994a Model reduction via least-squares Padé simplification of squared-magnitude functions. *Int. J. Systems Sci.*, **25:** 1191-1204.

Aguirre, L.A. 1994b Partial least-squares Padé reduction with exact retention of poles and zeros. *Int. J. Systems Sci.*, **25:** 2377-2391.

Aguirre, L.A. 1994c Computer-aided analysis and design of control systems using model approximation techniques. *Comp. Methods Appl. Mech. Eng.*, **114:** 273-294.

Aguirre, L.A. 1995 Algorithm for extended least-squares model reduction. *Electron. Lett.*, **31:** 1957-1958.

Aguirre, L.A. and Mendes, E.M.A.M. 1995 The least-squares Padé method for model simplification of multivariable systems. *Int. J. Systems Sci.*, **26:** 819-839.

Badyopadhyay, B. and Lamba, S.S. 1987 Time-domain Padé approximation and modal-Padé method for multivariable systems. *IEEE Trans. Circuits & Sys.*, **34:** 91-94.

Bultheel, A. and Van Barel, M. 1986 Padé techniques for model reduction in linear system theory: a survey. *J. Comp. and Appl. Math.*, **14:** 401-438.

Chen, C.F. and Chan, H.W. 1985 A note on Jury's stability test and Kalman-Bertram's Liapunov function. *Proc. IEEE*, **73:** 160-161.

Chen, T.C., Chang, C.Y. and Han, K.W. 1979 Reduction of transfer functions by the stability equation method. *J. Franklin Inst.*, **308:** 389-404.

Chen, T.C., Chang, C.Y. and Han, K.W. 1980 Model reduction using the stability equation method and the continued-fraction method. *Int. J. Control*, **32:** 81-94.

Chen, C.F. and Shieh, L.S. 1968 A novel approach to linear model simplification. *Int. J. Control*, **8:** 561-570.

Chuang, S.C. 1970 Application of continued-fraction method for modelling transfer functions to give more accurate initial transient response. *Electron. Lett.*, **6:** 861-863.

Conte, S.D. 1965 Elementary Numerical Analysis: An Algorithmic Approach. *McGraw-Hill Series in Information Processing and Computers*, McGraw-Hill, New York.

Davidson, A.M. and Lucas, T.N. 1974 Linear system reduction by continued-fraction expansion about a general point. *Electron. Lett.*, **10:** 271-273.

Davidson, A.M. and Lucas, T.N. 1976 Linear system reduction using Schwarz canonical form. *Electron. Lett.*, **12:** 324.

Davidson, A.M. and Walters, I.R. 1988 Linear system reduction using approximate moment-matching. *IEE Proc. D.*, **135:** 73-78.

Davison, E.J. 1966 A method for simplifying linear dynamic systems. *IEEE Trans. Autom. Control*, **11:** 93-101.

Fernando, K.V. and Nicholson, H. 1983 Reciprocal transformations in balanced model order reduction. *IEE Proc. Pt. D*, **130:** 359-362.

Green, M. and Limebeer, D.J.N. 1995 Linear robust control. *Prentice-Hall Information and System Science Series*, Prentice-Hall, Eaglewood Cliffs, New Jersey.

Gutman, P., Mannerfelt, C.F. and Molander, P. 1982 Contributions to the model reduction problem. *IEEE Trans. Autom. Control*, **27:** 454-455.

Hutton, M.F. Ph.D. thesis, Polytechnic Institute of New York, Brooklyn, N.Y.

Hutton, M.F. and Friedland, B. 1975 Routh approximations for reducing order of linear, time-invariant systems. *IEEE Trans. Automat. Ctrl.*, **20:** 329-337.

Hwang, C., Hwang, J.-H. and Guo, T.-Y. 1995 Multifrequency Routh approximants for linear systems. *IEE Proc. Contol Theory Appl.*, **142:** 351-358.

Hwang, C. and Lee, Y.C. 1989 Multifrequency Padé approximationvvia Jordan continued-fraction expansion. *IEEE Trans. Autom. Control*, **34:** 444-446.

Hwang, C. and Suen, C. 1986 Stable simplification of z-transfer functions by squared magnitude continued-fraction expansion. *J. Franklin Inst.*, **322:** 1-12.

Jacobs O.L.R. 1993 Introduction to Control Theory, 2nd edition. *Oxford University Press.*

Jamshidi, M. 1983 Large-scale systems: modeling and control. *North-Holland Series in System Science and Engineering*, **9:** Elsevier Science Publishing Co., Inc., New York 10017.

Katsube, Y., Horiguchi, K. and Hamada N. 1985 System reduction by continued-fraction expansion about $s = j\omega_i$. *Electron. Lett.*, **21:** 678-680.

Krishnamurthi, V. and Seshadri, V. 1976 A simple and direct method of reducing order of linear systems using Routh approximations in the frequency domain. *IEEE Trans. Automat. Ctrl.*, **21:** 797-799.

Krishnamurthi, V. and Seshadri, V. 1978 Model reduction using the Routh stability criterion. *IEEE Trans. Automat. Ctrl.*, **23:** 730-731.

Lal, M., Mitra, R. and Jain, A.M. 1975 On Schwarz canonical form for large system simplification. *IEEE Trans.*, **AC-20:** 262-263.

Lalonde R.J. 1992a The calculation of reduced order linear models from input/output data of high order nonlinear systems. *Ph.D Thesis, University of Akron, Ohio, USA.*

Lalonde R.J., Hartley T.T. and De Abreu-Garcia, J.A. 1992b Least-squares model reduction. *J. Franklin Inst.*, **329:** 215-240.

Lucas, T.N. 1978 Ph.D. thesis, U.W.I.S.T., Cardiff, Wales.

Lucas, T.N. 1983a Factor division: a useful algorithm for model reduction. *Proc IEE*, **130:** 362-364.

Lucas, T.N. 1983b Continued-fraction algorithm for biased model reduction. *Electron. Lett.*, **19:** 444-445.

Lucas, T.N. 1983c Efficient algorithm for reduction by continued-fraction expansion about $s = 0$ and $s = a$. *Electron. Lett.*, **19:** 991- 993.

Lucas, T.N. 1986 Continued-fraction expansion about two or more points: a flexible approach to linear system reduction. *J. Franklin Inst.*, **321:** 49-60.

Lucas, T.N. 1989 New results on relationships between multipoint Padé approximation and stability preserving methods in model reduction. *Int. J. Systems Sci.*, **20:** 1267-1274.

Lucas, T.N. 1992 A tabular approach to the stability equation method. *J. Franklin Inst.*, **329:** 171-180.

Lucas, T.N. 1993a New matrix method for multipoint Padé approximation of transfer functions. *Int. J. Systems Sci.*, **24:** 809-818.

Lucas, T.N. 1993b Optimal model reduction by multipoint Padé approximation. *J. Franklin Inst.*, **330:** 79-93.

Lucas, T.N. 1993c Extension of matrix method for complete multipoint Padé approximation. *Electron. Lett.*, **29:** 1805-1806.

Lucas, T.N. 1993d Optimal discrete model reduction by multipoint Padé approximation. *J. Franklin Inst.*, **330:** 855-867.

Lucas, T.N. 1994 Suboptimal model reduction by multipoint Padé approximation. *Proc. I. Mech. E. Part 1: J. Sys. Control Eng.*, **208:** 131-134.

Lucas, T.N. and Beat, I.F. 1990 Model reduction by least-squares moment matching. *Electron. Lett.*, **26:** 1213-1215.

Lucas, T.N. and Davidson, A.M. 1983 Frequency-domain reduction of linear systems using Schwarz approximation. *Int. J. Control*, **37:** 1167-1178.

Lucas, T.N. and Munro, A.R. 1991 Model reduction by generalised least-squares method. *Electron. Lett.*, **27:** 1383-1384.

Lucas, T.N. and Smith, I.D. 1995 Least-squares Padé reduction: A nonuniqueness property. *Electron. Lett.*, **31:** 1640-1642.

Lucas, T.N. and Smith I.D. 1998 Discrete-time least-squares Padé order reduction: A stability preserving method. *Proc. I. Mech. E. Part I, J. Sys. & Cont. Eng.*, **212:** 49-56.

MacFarlane, A.G.J. 1974 Relationships between recent developments in linear control theory and classical design techniques. *IFAC Symposium on multivariable technological systems*, Manchester, U.K.

Moore, B.C. 1981 Principal component analysis in linear systems: controllability, observability and model reduction. *IEEE Trans. Autom. Control*, **AC-26:** 17-32.

Pal, J. 1986 An algorithmic method for the simplification of linear dynamic scalar systems. *Int. J. Control*, **43:** 257-269.

Parthasarathy, R. and Jayasimha, K.N. 1982 System reduction using stability-equation method and modified Cauer continued fraction. *Proc. IEEE*, **AC-70:** 1234-1235.

Pernebo, L. and Silverman, L.M. 1982 Model reduction via balanced state space representations. *IEEE Trans. Aut. Control*, **AC-27:** 382-387.

Rosenbrock, H.H. 1970 State-space and Multivariable theory. *Nelson*, London

Rao, A.S., Lamba, S.S., and Rao, S.V. 1978 On simplification of unstable systems using Routh approximation technique. *IEEE Trans. Automat. Ctrl.*, **23:** 943-944.

Sarasu, J. and Parthasarathy, R. 1979 System reduction by Routh approximation and modified Cauer continued fraction. *Electron. Lett.*, **15:** 691-692.

Shamash, Y. 1973 Ph.D. Thesis, Imperial College, University of London.

Shamash, Y. 1974a Stable reduced order models using Padé approximation. *IEEE Trans. Automatic Control*, 615-616.

Shamash, Y. 1974b Continued-fraction methods for the reduction of discrete-time systems. *Int. J. Control*, **20:** 267-275.

Shamash, Y. 1975a Linear system reduction using Padé approximation to allow retention of dominant modes. *Int. J. Control*, **21:** 257-272.

Shamash, Y. 1975b Model reduction using the Routh stability criterion and the Padé approximation technique. *Int. J. Control*, **21:** 475-484.

Shamash, Y. 1975c Multivariable system reduction via modal methods and Padé approximation. *IEEE Trans. Automat. Ctrl.*, Technical Notes and Correspondence : 815-817.

Shamash, Y. 1976 Continued fraction methods for the reduction of constant-linear multivariable systems. *Int. J. Systems Sci.*, **7:** 743-758.

Shamash, Y. 1978 Routh approximations using the Schwarz canonical form. *IEEE Trans. Automat. Ctrl.*, **23:** 940-941.

Shamash, Y. and Feinmesser, D. 1978 Reduction of discrete time systems using a modified Routh array. *Int. J. Systems Sci.*, **9:** 53-64.

Shieh, L.S. and Wei, Y.S. 1975 A mixed method for multivariable system reduction. *IEEE Trans. Automat. Ctrl.*, Technical Notes and Correspondence : 429-432.

Shoji, F.F., Abe, K. and Takeda, H. 1985 Model reduction for a class of linear dynamic systems. *J. Franklin Inst.*

Sinha, P.K. 1984 Multivariable Control. *Electrical Engineering and Electronics Series/19*, Marcel and Dekker.

Smith, I.D. and Lucas, T.N. 1995 Least-squares moment matching reduction methods. *Electron. Lett.*, **31:** 929-930.

Smith, I.D. and Lucas, T.N. 1996 A unifying theory of least-squares Padé model reduction methods. *Proc. I. Mech. E. Part I: J. Sys. Control Eng.*, **210:** 95-102.

Taiwo, O. and Krebs, V. 1993 Generalized moment method for the order reduction of multivariable systems. *J. Franklin Inst.*, **330:** 641-649.

Taiwo, O. and Krebs, V. 1995 Multivariable system simplification using moment matching and optimisation. *IEE Proc. Control Theory Appl.*, **142:** 103-110.

Therapos, C.P. 1983 Stability equation method to reduce the order of fast oscillating systems. *Electron. Lett.*, **19:** 183-184.

Therapos, C.P. and Diamessis J.E. 1984 Sampling method for linear system reduction. *J. Franklin Inst.*, **317:** 359-371.

Towill, D.R. 1972 Low order modelling techniques: Tools or Toys, *Computer Aided Design and Control, IEE*, 206-212

Wall, H. S. 1948 Analytic Theory of Continued Fractions. *Van Nostrand Company Ltd.*, Princeton, New Jersey.

Xiheng, H. 1987 FF-Padé method of model reduction in frequency domain. *IEEE Trans. Autom. Control*, **32:** 243-246.

Yahagi, T. 1980 On the simplification of transfer functions of linear dynamical systems. *J. Dyn. Sys., Measurement and Control*, **102:** 7-12.